

**NASA Contractor Report 182008**

**AN OVERLAPPED GRID METHOD FOR  
MULTIGRID, FINITE VOLUME/DIFFERENCE  
FLOW SOLVERS - MaGGiE**

**Oktay Baysal and Victor R. Lessard**

**OLD DOMINION UNIVERSITY RESEARCH FOUNDATION  
Norfolk, Virginia**

**Grant NAG1-664  
February 1990**



National Aeronautics and  
Space Administration

**Langley Research Center**  
Hampton, Virginia 23665-5225

(NASA-CR-182008) AN OVERLAPPED GRID METHOD  
FOR MULTIGRID, FINITE VOLUME/DIFFERENCE FLOW  
SOLVERS: MaGGiE Report, 1987 - 1989 (Old  
Dominion Univ.) 111 p

CSCL 01A

N90-20953

Unclass

G3/02 0277010



## ABSTRACT

### AN OVERLAPPED GRID METHOD FOR MULTIGRID, FINITE VOLUME/DIFFERENCE FLOW SOLVERS - MaGGiE

Computing the flow fields about three-dimensional complex configurations accurately becomes a difficult task, if it is attempted to generate a single, body fitted grid with proper clustering. The domain decomposition methods, which divide the computational domain into less complex subdomains, are extensively used to decrease the grid generation workload. A domain decomposition technique also allows the use of different solution methods for different subdomains. The objective of this work is to develop a domain decomposition method via overlapping/embedding the component grids, which is to be used by upwind, multigrid, finite volume solution algorithms. A computer code, given the name MaGGiE, (Multi-Geometry Grid Embetter), is developed to meet this objective. MaGGiE takes independently generated component grids as input, and automatically constructs the composite mesh and interpolation data, which can be used by the finite volume solution methods with or without multigrid convergence acceleration. Six demonstrative examples, showing various aspects of the overlap technique are presented and discussed. These cases are: the grid of a blunt-nose cylinder, (BNC), embedded within a Cartesian farfield, with finest level and multi-level grid connections, where the flow Mach number is 1.6, and the angle of attack is  $32^\circ$ ; the grid of BNC is overlapped within a farfield mesh of similar topology for the same flow

conditions as the previous case; an ogive-nose cylinder, (ONC), in the proximity of a flat plate, where the flow Mach number is 2.86; a cylindrical store model connected to an L-shaped sting, embedded within a Cartesian farfield, where the flow Mach number is 1.65; a different cylindrical store model with fins and a curved sting in the proximity of a cavity. These cases are used for developing the procedure for overlapping grids of different topologies, and to evaluate the grid connection and interpolation data for finite volume calculations on a composite mesh. The flow solutions are obtained for all the cases, except the one which involves the cavity. Time fluxes are transferred between mesh interfaces using a trilinear interpolation procedure. Conservation losses are minimal at the interfaces using this method. The multigrid solution algorithm, using the coarser grid connections, improves the convergence time history as compared to the solution on composite mesh without multigriding.

## TABLE OF CONTENTS

	Page
ABSTRACT .....	i
TABLE OF CONTENTS .....	iii
LIST OF SYMBOLS .....	v
Chapter	
1. INTRODUCTION .....	1
1.1 Rationale .....	1
1.2 Literature Survey .....	5
1.3 Present Work .....	9
2. GOVERNING EQUATIONS OF FLUID FLOW .....	10
3. BASELINE SOLUTION ALGORITHM ON A SINGLE DOMAIN .....	14
3.1 Finite Volume Discretization .....	14
3.2 Upwind Differencing .....	15
3.3 Roe Flux-Difference-Splitting .....	15
3.4 Multigrid Method .....	17
3.5 Initial and Boundary Conditions .....	19
4. GRID OVERLAPPING METHOD .....	23
4.1 Grid Generation .....	23

4.2 Overlapping Algorithm .....	24
4.2.1 Hole Boundary .....	26
4.2.2 Outer Boundary .....	31
4.2.3 Output Format .....	32
4.2.4 Composite Grids for Multigrid Method .....	33
4.3 Overlapped Region in a Composite Mesh .....	35
4.4 Inter-Subdomain Conservation .....	35
4.5 Global Accuracy .....	39
4.6 Modified Solution Algorithm .....	39
4.7 Procedure for Solution Algorithm.....	42
 5. APPLICATIONS OF GRID OVERLAPPING .....	 44
5.1 Blunt-Nose Cylinder in a Cartesian Grid (Single level) .....	45
5.2 Blunt-Nose Cylinder in a Cartesian Grid (Multi-level).....	48
5.3 Blunt-Nose Cylinder with two Overlapped C-O grids .....	50
5.4 Ogive-Nose Cylinder Near a Flat Plate .....	52
5.5 Store Model with L-Sting in Cartesian Grid .....	54
5.6 Store Model with Fins and Curved Sting Near a Cavity .....	58
5.7 Comparisons and Comments .....	60
 6. CONCLUSIONS .....	 61
 REFERENCES .....	 64
FIGURES .....	69
APPENDIX A: CALCULATIONS OF NORMAL VECTORS .....	98
APPENDIX B: JACOBIAN OF ISOPARAMETRIC TRANSFORMATION .....	101

## LIST OF SYMBOLS

$a$	speed of sound
$C_v$	specific heat at constant volume
$c$	Sutherland constant
$D$	diameter
$E$	total energy per unit mass
$e$	internal energy per unit mass, or exponential function
$F, G, H$	inviscid flux vectors
$F_v, G_v, H_v$	viscous flux vectors
$G_n$	coarse grid level
$G_l$	composite grid level
$J$	Jacobian matrix
$k$	thermal conductivity
$M$	Mach number, or coarse grid level
$n$	normal direction
$P$	static pressure
$p$	order of differential equation
$Q$	time flux vector
$q$	heat flux
$R$	universal gas constant, or Riemann invariants, or residual

$r$	order of accuracy for spatial discretization
$Re$	Reynolds number
$S$	surface
$s$	arc length
$T$	temperature
$t$	time
$U, V, W$	contravariant velocity components
$u, v, w$	Cartesian velocity components
$X, Y, Z$	Cartesian coordinates

# **GREEK**

$\alpha$	interpolation coefficient
$\delta$	Kronecker delta
$\tau$	stress
$\xi, \eta, \zeta$	curvilinear coordinates
$\mu$	molecular viscosity
$\lambda$	second coefficient of viscosity
$\rho$	density
$\Sigma$	summation
$\gamma$	ratio of specific heats
$\Lambda$	diagonal matrix
$\Delta$	change

# **SUBSCRIPTS**

$\infty$	freestream
----------	------------



$i, j, k$	indices for $\xi, \eta, \zeta$ directions
$L$	left
$R$	right
$v$	viscous

## **SUPERSCRIPTS**

$\rightarrow$	vector
$(\bar{\phantom{x}})$	quantity in locally orthogonal coordinates
$(^{\wedge})$	normalized quantity
$b$	values on a boundary
$n$	iteration level

## **OPERATORS**

$\delta$	upwind difference
$\delta^2$	central difference



## Chapter 1

# INTRODUCTION

### 1.1 Rationale

Computational fluid dynamics (CFD) plays a dominant role in the aerospace field because of the realization that CFD is an effective design tool which complements and goes beyond experimental tests. Because of the rapid development of computational fluid dynamics in the last decade, efficient solvers, capable of solving the partial differential equations of fluid motion by finite-difference (FD), finite-volume (FV) and finite-element (FE) techniques, have evolved. Validation of these codes have caused the important merging of the computational and experimental disciplines. Coinciding with the theoretical advancements is the continuing improvements of high speed and large memory digital computers with vectorization and parallel processing capabilities. The CFD community has lead the push for the state of the art supercomputer technology and scientific workstations. With the continuing advancements in computer hardware and software, it has become practical to solve three dimensional complex flow domains, which were previously thought to be beyond the reach of the computational fluid dynamics.

The term complex flow field can be defined as any physical domain in which there are high flow field gradients, and a single or multiple bodies of nonsmooth, multiple joint or disjoint geometries. A few examples of complex flow domains are the flow around an aircraft, the flow between a wing and a

store, the flow between a store and a cavity, the flow between a wing and a nacelle, etc.. Due to the complexities of these real bodies, it is a formidable task to generate global, body fitted grids with, requisite smoothness and cell clustering in high-gradient regions that are supportive to the new sophisticated flow solvers. The body-fitted or boundary conforming curvilinear grids are desirable, because they provide a basic advantage of implementing the surface boundary conditions accurately. Also, a proper surface oriented coordinate system enables coordinate-related approximations to the equations of motions for arbitrary complex geometries. It becomes more difficult to locally control the orthogonality, volume variations, cell aspect ratios, and other grid measures, which affect the accuracy of the solution as the geometric complexity increases [1]\*. To reduce the grid generation task about complicated geometries, several approaches, such as, domain decomposition and unstructured grids, have been investigated by researchers.

The unstructured grid approach discretize the flow field by triangular elements, or tetrahedrons, with nodes placed at the vertices. Discretizing the flow by such elements gives flexibilities in grid generation about complex geometries. The unstructured grid method is primarily used with finite-element techniques. One disadvantage of unstructured grids is the extra amount of storage needed for the grid structure order number. It is a rather difficult task to generate unstructured grids in the close proximity of a solid surface, where example, clustering is needed for viscous solutions. Also, since FDM and FVM are computationally more efficient when compared to FEM for the Navier Stokes equations, unstructured grids may become less desirable. However, a hybrid grid system composed of unstructured and structured grids

---

\* The numbers in the braces indicate references.

developed by Nakahashi et al. [2] offers a promising approach to complex flow domains.

The domain decomposition techniques are of primary interest in this study. The two principle elements of the domain decomposition method, (DDM), are the subdivision of the computational domain and the communication among the subdomains. The DDM divides the flow region into simpler subdomains within which grids are independently or semi-independently generated using existing grid generation schemes. Some current grid generation methods are the algebraic method, the conformal-mapping method, the differential-equations method. An advantage of the DDM is that the flow regions requiring grid clustering can be isolated into different subdomains. In addition, the decomposition method enables the use of different partial differential equations and solution methods for different subdomains. This is particularly beneficial when using subdomains near and far away from a body. The Navier Stokes equations can be used to investigate the domain near the body and the Euler equations can be used in the farfield. This may result in a saving of computer time. Another advantage of the DDM is the domain block-processing scheme where only data corresponding to particular subdomain is required to reside in the main memory of the computer at one particular time. Thus, the block-processing technique ideally permits the use of unlimited global grid sizes. The second and the most critical element of the DDM is the communication between grid domain. Communication, or data transference, between domain boundaries are accomplished by some type of interpolation method of either nonconservative or conservative nature.

Zonal method (or grid patching) and grid overlapping/embedding are the two most common domain decomposition techniques used by current researchers. Zonal method incorporates the techniques of patching grids

together along common boundaries or surfaces to create a global grid. The main disadvantage of using grid patching is that the patched zones of connecting grids have to lie on the same surface. This characteristic of grid patching increases the complexities of grid generation for each subdomain. Another discouraging feature of grid patching is the loss of conservation across zones of high curvature.

An alternate domain decomposition method is the grid overlapping. Grid overlapping entails dividing the flow domain into regions that overlap or share common physical and computational space. Within the overlap region, the grids communicate through data transference by an interpolation procedure. Grid embedding schemes allow the subdomains to be non-disjoint so that one mesh may be embedded completely or partially within another. This procedure permits each subdomain to be meshed independently with no requirements of continuous grid lines across boundaries. Because each subdomain grid is independent of another, grid generation task is greatly reduced for complicated flow regions. Each subdomain mesh can be created using different grid generation techniques suitable for that particular domain. This is specially beneficial for subdomains which require high grid densities. Again re-emphasizing, the advantage of grid overlapping/embedding techniques is that subdomain grids of different topologies can be connected in many different ways to encompass the entire flow field. This is the driving force behind the current thesis work on the grid overlapping/embedding method.

There are several drawbacks of using the embedding method, but most problems can be partially or completely alleviated. The disadvantages are the following: (i) the technique requires an overlap region between subdomains which may not always be feasible, (ii) the accuracy of boundary data

transference depends on the interpolation procedure, whether it is conservative or nonconservative, and (iii) the accuracy and convergence speed of the solution indirectly depend on the degree of overlapping of the grids relative to the size of the subdomains.

## 1.2 Literature Survey

In 1982, Hennesius et al. [3] developed a zoning technique for the Euler equations within the framework of an implicit numerical scheme for one- and two-dimensional equations. Their scheme required continuity of the mesh in point and slope near the interfacing region. They concluded that proper flux balancing was necessary, when zonal boundaries are present near converged shock locations or in large gradient regions. In 1984, Rai [4] used a conservative treatment of zonal boundaries for solving the Euler equations. The scheme of Rai did not require continuity for mesh in point and slope at the zonal boundaries. The capability of having grid discontinuities between zones enhances the zonal method for complex flow domains. However, zonal boundaries with moderate curvature were shown to lose conservation. In 1986 Hennesius et al. [5] were one of the first to develop a three dimensional conservative boundary scheme for patched grids, applicable in generalized coordinates, for arbitrary point distribution on a planar surface. It was shown that the three-dimensional zonal method simplifies the grid generation about complex configurations, by its application to the computation of flow about a wing-canard combination, using two interfacing patched grids.

Kathong [6] studied the feasibility of the conservative Ramshaw [7] grid patching procedure for applications to realistic three dimensional aerodynamic configurations. The Ramshaw method has no restrictions on grid slope or density across zonal boundaries. The results concluded that global

conservation can be maintained across grid interfaces for complex configurations.

In 1989, Thomas et al. [8] developed a patched-grid algorithm for the analysis of complex configurations using an implicit upwind-biased Navier-Stokes solver. The patched-grid application was directed towards the F-18 aircraft at subsonic, high angle of attack conditions. A difference between spatial-flux and time-flux conservation across zonal interfaces were compared. It was noted that there was little difference in the results between the spatial-flux and time flux conservation approach. The time flux approach (interpolating to the cell center of one grid, assuming a linear variation of the flux within cells of the other grid) was considered more flexible and lends itself to more complicated conditions, such as, overlapped and embedded grids. Thomas et al. proposed a long term objective to develop an automatic, generic domain decomposition method to handle zonal, overlapped, and embedded grids with the only constraint on the grids being that the grids encompass the entire flow domain.

Another form of grid patching is a domain hybrid method developed by Nakahashi et al. [9]. The hybrid method divides a complex domain into regions of structured and unstructured grids as briefly discussed previously. Structured grids are used in the viscous flow regions, and are patched together using unstructured grids. With this technique both computational efficiency of FDM or FVM in the structured region and that of FEM in the geometrical flexible region of unstructured grids can be obtained.

Earlier work in grid overlapping was done for finite difference flow solvers. In 1981, Atta [10] developed a method for constructing a two dimensional grid system for solving the transonic flow field about complex configurations with multiple components. His test model was a two component



configuration that consisted of an airfoil embedded in rectangular boundaries. The results showed that the accuracy and convergence speed of an implicit approximate factorization scheme depended on the extent of the overlap region and the size of each subdomain. In 1982, Atta et al. [11] extended the two dimensional overlap scheme to three dimensions for the case of an isolated wing and a wing/pylon/nacelle configuration. The transfer of information between grids within the overlap regions was done by a trivariate interpolation polynomial based on a linear Taylor series expansion. A fully implicit, approximate factorization scheme was used for finite differenced, full potential equations..

Benek et al. [12-14] developed a generic grid overlapping/embedding procedure known as the "chimera scheme", for in two- and three-dimensional, and finite difference solutions of the Euler equations. The chimera scheme involves the automatic connection of multiple, overset grids, and the use of different solution procedures for different subdomain grids. The chimera scheme is one in which a major grid covers the entire flow region, and minor grids are then overset on the major grid so as to resolve secondary features of the configuration, such as, flaps, nacelles or stores, etc. The minor grids are fully or partially overlapped without, requiring the mesh boundaries to join in any special way. The minor grids create holes in the major grid, which are excluded from the solution of the major grid. Communications between the major and minor grids occur within the overlap regions. The chimera method was successfully demonstrated on several geometries for inviscid flow. In 1987, Benek et al. [15] extended the chimera grid embedding scheme with applications to viscous flows. They developed generalization of rules for constructing subdomains, and added thin-layer Navier-Stokes

equations to the model. These extensions to the chimera scheme were applied to a single axisymmetric body and a three-body configuration.

In 1987, Suhs [16] used the chimera grid scheme in the computation of a three dimensional cavity flow at subsonic and supersonic Mach numbers. The cavity flow was calculated using an implicit, finite difference Navier-Stokes code with thin-layer approximations. Although the thin layer approximations are inappropriate for the unsteady cavity flow, Suhs showed the versatility of the chimera scheme for simplifying a complex flow domain into simpler subdomains of Cartesian grids.

In 1989, Dougherty et al. [17,18] applied the chimera grid scheme to three-dimensional transonic store separation. Inviscid finite difference calculations were carried out for a minor store mesh moving with respect to the major mesh. The results indicate that allowing one mesh to move with respect to another does not adversely effect the time accuracy of an unsteady flow. The results of the moving mesh scenario shows the importance of overlapped/embedded schemes. The flow around multiple bodies moving relative to each other cannot be solved using single, patched or unstructured grids.

Recently, Chesshire et al. [19,20] have developed a technique for the generation of curvilinear composite overlap grids and the numerical solution of partial differential equations on them. Continuity conditions through interpolations are imposed at the overlap boundaries. Their grid construction program, CMPGRD, is used to create composite, two dimensional, and very recently three dimensional, grids with any number of component grids, for finite difference and finite volume computations. The CMPGRD program can generate a composite grid which can be used for second or higher order spatial discretizations with appropriate higher order interpolation. However,

the higher order interpolations require a greater overlap region between subdomains and considerably more calculations. CMPGRD program is also designed to automatically generate the sequence of coarser grids needed in a multigrid algorithm flow solver.

### **1.3 Present Work**

The objective of the present work is to develop an overlapping procedure for multiple grids around complex flow configurations, which is to be used by a multigrid, finite volume solution algorithm, and to apply this method to several complex flow problems. The flow problems investigated are as follows: blunt-nose cylinder embedded within two different farfield grid topologies, with the flow at an angle of attack of  $32^\circ$ ; supersonic flow past an ogive-nose cylinder in the proximity of a flat plate; supersonic flow past a cylindrical store model connected to an L-shaped sting; and a complex configuration of a cylindrical store model with fins and curved sting in proximity of a cavity.

This report is divided into chapters of logical sequence. Chapter 2 conveys the governing equations of fluid motion. The baseline solution algorithm on a single domain is given in Chap. 3. Chapter 4 describes the grid overlapping method for solvers with and without multigriding, after a brief introduction on grid generation for subdomain grids. Grid interface conservation and global accuracy are also discussed in this chapter. The flow solver methodology for multiple subdomains, including modified solution algorithm and run procedure, are also given in Chap. 4. Chapter 5 covers the grid overlapping applications, a summary of comparisons and comments. The concluding remarks and appropriate suggestions for further investigations in this area are presented in Chap. 6.

## Chapter 2

### GOVERNING EQUATIONS OF FLUID FLOW

The governing equations are the three-dimensional, time dependent, complete, Reynolds-averaged, Navier-Stokes equations, written in conservative form and generalized curvilinear coordinates,  $\xi, \eta, \zeta$ :

$$\frac{\partial Q}{\partial t} + \frac{\partial (F - F_v)}{\partial \xi} + \frac{\partial (G - G_v)}{\partial \eta} + \frac{\partial (H - H_v)}{\partial \zeta} = 0 \quad (2.1)$$

Written in a more compact indicial form the equation becomes

$$\frac{\partial Q}{\partial t} + \frac{\partial (F^i - F_v^i)}{\partial \xi^i} = 0 \quad (2.2)$$

where  $i = 1, 2, 3$ . The  $Q$  vector of conserved variables is

$$Q = \left[ \rho, \rho u, \rho v, \rho w, \rho E \right]^T / J, \quad (2.3)$$

$F^i$  are the inviscid flux vectors,

$$F^i = \frac{1}{J} \begin{bmatrix} \rho U_i \\ \rho u U_i + \xi_x^i p \\ \rho v U_i + \xi_y^i p \\ \rho w U_i + \xi_z^i p \\ (E + p) U_i + \xi_p^i p \end{bmatrix} \quad (2.4)$$

and  $F_v^i$  are the viscous flux vectors,

$$F_v^i = \frac{1}{J} \begin{bmatrix} 0 \\ \xi_k^i \tau_{k1} \\ \xi_k^i \tau_{k2} \\ \xi_k^i \tau_{k3} \\ \xi_k^i (u \tau_{k1} + v \tau_{k2} + w \tau_{k3} - q_k) \end{bmatrix} \quad (2.5)$$

The contravariant velocity components are defined by

$$\begin{aligned} U_1 = U &= \xi_x u + \xi_y v + \xi_z w + \xi_t \\ U_2 = V &= \eta_x u + \eta_y v + \eta_z w + \eta_t \\ U_3 = W &= \zeta_x u + \zeta_y v + \zeta_z w + \zeta_t \end{aligned} \quad (2.6)$$

and the transformation Jacobian is defined by

$$\begin{aligned} J &= \frac{\partial(\xi, \eta, \zeta)}{\partial(x, y, z)} \\ &= x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi \\ &\quad - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi \end{aligned} \quad (2.7)$$

A geometrical interpretation of the metric terms can be made using a control volume approach. The ratio of a metric derivative to the transformation Jacobian for a given cell, for example  $\left(\frac{\xi_x}{J}\right)$ , is taken to be the appropriate projected area of a cell face. The reciprocal of the Jacobian is taken to be the cell volume. This approach ensures the geometric conservation law to be compatible with the finite volume formulation.

The shear stress and heat flux terms used in the above equations are given by

$$\begin{aligned}\tau_{k1} &= \mu \left( \xi_1^m \frac{\partial u_k}{\partial \xi^m} + \xi_k^m \frac{\partial u_1}{\partial \xi^m} - \frac{2}{3} \delta_{k1} \xi_n^m \frac{\partial u_n}{\partial \xi^m} \right) \\ \tau_{k2} &= \mu \left( \xi_2^m \frac{\partial u_k}{\partial \xi^m} + \xi_k^m \frac{\partial u_2}{\partial \xi^m} - \frac{2}{3} \delta_{k2} \xi_n^m \frac{\partial u_n}{\partial \xi^m} \right) \\ \tau_{k3} &= \mu \left( \xi_3^m \frac{\partial u_k}{\partial \xi^m} + \xi_k^m \frac{\partial u_3}{\partial \xi^m} - \frac{2}{3} \delta_{k3} \xi_n^m \frac{\partial u_n}{\partial \xi^m} \right) \\ q_k &= k \xi_k^m \frac{\partial T}{\partial \xi^m}\end{aligned}\tag{2.8}$$

where k,n and m are dummy variables and  $\xi_1=\xi_x$ ,  $\xi_2=\xi_y$ ,  $\xi_3=\xi_z$ .

The total energy, E, and the internal energy, e, are given by:

$$\begin{aligned}E &= e + \frac{1}{2} (u^2 + v^2 + w^2) \\ e &= C_v T\end{aligned}\tag{2.9}$$

The perfect gas law,

$$P = \rho RT\tag{2.10}$$

and the Sutherland's molecular viscosity law,

$$\mu = T^{3/2} \left( \frac{1 + c/T_{\infty}}{T + c/T_{\infty}} \right) \quad (2.11)$$

with  $c$  being the Sutherland constant, and Stokes' hypothesis for bulk viscosity,

$$\lambda + 2\mu/3 = 0 \quad (2.12)$$

completes the closure of the system of governing equations. Reynolds stresses are modeled by the standard Baldwin-Lomax algebraic turbulence model. Further details of this formulation are given in [21,22].

## Chapter 3

### BASELINE SOLUTION ALGORITHM ON A SINGLE DOMAIN

The solution algorithm for multiple subdomains is based on an implicit, upwind, finite-volume algorithm for a single domain. The solution algorithm for the multiple subdomain algorithm is discussed in Chap. 4.

#### 3.1 Finite Volume Discretization

Finite volume differencing is formulated by integrating the conservation equations over a stationary control volume,

$$\frac{\partial}{\partial t} \int \int \int_V Q dV + \int \int_S \vec{F} \cdot \vec{n} dS = 0 \quad (3.1)$$

where the flux vector  $\vec{F}$  is defined as

$$\vec{F} = (F - F_v)\vec{i} + (G - G_v)\vec{j} + (H - H_v)\vec{k} \quad (3.2)$$

and

$$\vec{n} = n_x\vec{i} + n_y\vec{j} + n_z\vec{k} \quad (3.3)$$

is the unit normal vector pointing outward from the surface  $S$ , bounding the volume  $V$ . The direct discretization of the integral form ensures that mass, momentum and energy are conserved at discrete levels. The conserved variables,  $Q$ , are evaluated at cell centers and the fluxes,  $F^i$ , are evaluated at cell faces. The advantages of the finite volume formulation is that it remains valid in the presence of discontinuities in the flow, such as shocks, and that it



it is tolerant to grid singularities because the flow equations are balanced over each cell of the grid.

### 3.2 Upwind Differencing

The time-dependent Euler equations form a system of hyperbolic equations, and upwind differencing [23-26] models the characteristic nature of these equations in that information at each grid cell is obtained from directions dictated by characteristic theory. Upwind methods have the advantage of being naturally dissipative, unlike central differencing methods in which artificial dissipation terms are generally needed to overcome oscillation or instabilities arising in regions of high gradients.

### 3.3 Roe Flux-Difference-Splitting

The upwind scheme used for the test cases is based on the Roe flux-difference-splitting. Roe flux-difference splitting [27] is used to construct the upwind differences for the convective and pressure terms. If an eigenvalue of a flux Jacobian vanishes, the corresponding eigenvalue of the dissipation matrix also vanishes. This leads to a one or two cell resolution of discontinuities such as shocks. The spatial derivatives are written conservatively as flux balances across a cell, for example,

$$\left( \frac{\partial F}{\partial \xi} \right)_i = \frac{(F_{i+1/2} - F_{i-1/2})}{(\xi_{i+1/2} - \xi_{i-1/2})} \quad (3.4)$$

where the subscript 'i' refers to a cell center and i+1/2 corresponds to a cell surface. The interface flux is determined from a state-variable interpolation and a locally one-dimensional model of wave interactions normal to the cell interfaces. The interface fluxes are exact solutions to an approximate Riemann problem,

$$F_{i+\frac{1}{2}} = \frac{1}{2} \left\{ F(Q_L) + F(Q_R) - |A| (Q_R - Q_L) \right\}_{i+\frac{1}{2}} \quad (3.5)$$

where  $Q_L$  and  $Q_R$  are the state variables to the left and right of the cell interfaces and

$$A = \frac{\partial F}{\partial Q} = T \Lambda T^{-1} = T (\Lambda^+ + \Lambda^-) T^{-1} \quad (3.6a)$$

$$\Lambda^\pm = (\Lambda \pm | \Lambda |) / 2 \quad (3.6b)$$

$$|A| = T | \Lambda | T^{-1} \quad (3.6c)$$

The diagonal matrix  $\Lambda$  is the matrix of eigenvalues of  $A$ , and  $T$ ,  $T^{-1}$  are the diagonalizing matrices. The state-variables  $Q_L$  and  $Q_R$  are formed from interpolations of primitive variables,  $(p, u, v, w, p)$ , which in effect determines the resulting accuracy of the scheme.

The accuracy of the scheme used is second-order spatial and first order temporal. Spatial approximate factorization and Euler backward time integration results in the solution through 5x5 block-tridiagonal matrix inversion in three directions. The delta form of the discretized Eq. (2.1) is given by

$$\begin{aligned} \left[ \frac{I}{J\Delta t} + \delta_\xi \frac{\partial F}{\partial Q} - \delta_\xi^2 \frac{\partial F_v}{\partial Q} \right] \Delta Q^* &= -R(Q^n) \\ \left[ \frac{I}{J\Delta t} + \delta_\eta \frac{\partial G}{\partial Q} - \delta_\eta^2 \frac{\partial G_v}{\partial Q} \right] \Delta Q^{**} &= \Delta Q^* \\ \left[ \frac{I}{J\Delta t} + \delta_\zeta \frac{\partial H}{\partial Q} - \delta_\zeta^2 \frac{\partial H_v}{\partial Q} \right] \Delta Q &= \Delta Q^{**} \end{aligned} \quad (3.7)$$

$$Q^{n+1} = Q^n + \Delta Q \quad (3.8)$$

In the preceding equation,  $R(Q^n)$  is the discretized representation of the spatial derivative terms in Eq. (2.1) evaluated at time level (n), and  $\delta, \delta^2$  denote upwind and central-difference operators, respectively.

Employing the approximate diagonal form of the spatial factors of Eq. (3.7), results in the saving of computational time for the initialization of flowfields. Each of the spatial factors is approximated with a diagonal inversion [28] as

$$\left[ \frac{1}{J\Delta t} + \delta_{\xi} \frac{\partial F}{\partial Q} \right] \Delta Q^* \equiv T \left[ \frac{1}{J\Delta t} + \delta_{\xi}^- \Lambda^+ + \delta_{\xi}^+ \Lambda^- \right] T^{-1} \Delta Q^* \quad (3.9)$$

Because of the repeated eigenvalues of  $\Lambda$ , only scalar diagonal inversions rather than block inversions are used in each direction.

### 3.4 Multigrid Method

Because of the additional computational work for the upwind flux-difference splitting method, it is desirable to accelerate the convergence rate, especially when steady-state solutions are sought. Accelerating the convergence rate becomes increasingly important as the mesh is refined, because the logarithm of the spectral radius for single-grid methods generally increase linearly with the mesh size, thus computing on fine grids is expensive. To accelerate the convergence rate, multigrid method is used with the upwind, finite volume scheme. The multigrid method damps the low-frequency errors which cause a slow asymptotic convergence rate by using a sequence of grids  $G_1, \dots, G_n$ . The grid  $G_1$  denotes the finest grid. Successively coarser grids can be formed by deleting every other mesh line on the next finer mesh. The high frequency errors are easily damped out on a given grid level while the low frequency errors remain. When transferring solution to a coarser grid, the low frequency errors of the previous finer grid become higher frequency errors due to the increase in cell sizes on the coarser grid.

In turn, the high frequency errors on the coarser grid are damped out using the same solution algorithm as on the previous finer grid [29-32].

A fixed V-cycle [23] strategy of solving from finest to coarsest then back to finest grid levels, is used where a predetermined number of iterations is performed at each grid level. The values of the dependent conserved variables (Q) and the residual (R) are passed from a finer grid to a coarser grid through volume-weighted restriction operators  $(I_i^{i+1})$  and  $(\hat{I}_i^{i+1})$ , respectively,

$$Q_{i+1} = (I_i^{i+1} Q_i) \quad (3.10a)$$

$$R_{i+1} = (\hat{I}_i^{i+1} R_i) \quad (3.10b)$$

$$I_i^{i+1} Q_i = \Sigma V Q / \Sigma V \quad (3.10c)$$

$$\hat{I}_i^{i+1} R_i = \Sigma R_i \quad (3.10d)$$

where  $Q_{i+1}$  and  $R_{i+1}$  are the next coarser level values obtained from the finest level values. The equations  $(I_i^{i+1} Q_i)$  and  $(\hat{I}_i^{i+1} R_i)$  are found from summations taken over all fine-grid cells that make up the coarse-grid cell, where V is the cell volumes. The entire solution is computed and stored on each grid level as opposed to only corrections being stored. This multigrid process is referred to as the full-approximation scheme (FAS).

Denoting the discrete analog of the operation in Eq. (2.1) by (L), and the relative truncation error by (E), the following equations are written,

$$L_{i+1}(Q_{i+1}) = R_{i+1} + E_{i+1} \quad (3.11)$$

$$E_{i+1} = L_{i+1}(I_i^{i+1} Q_i) - \hat{I}_i^{i+1} R_i \quad (3.11)$$

The solution on the coarse grid is driven by the fine grid and the relative truncation error (E) between the coarse and fine grids. During the cycling process, when the coarsest level is reached, computed corrections to  $\Delta Q$  values

at each level are prolonged to the next finer level through trilinear interpolations. One smoothing iteration is used to smooth the errors. The result of this multigrid strategy is that most of the work is carried out on the coarser grids where it is computationally cheaper due to the reduction of the number of grid points. Because of these advantages, it is worth incorporating the multigrid scheme within the multiblock, grid overlapping, solution algorithm. A discussion of the flow solver methodology for multiple subdomains using the multigrid technique will be discussed in Chap. 4.

### 3.5 Initial and Boundary Conditions

The accuracy of the solution to any physical flow is dependent on the initial and boundary conditions. The initial conditions usually correspond to the actual nature of the flow. The initial conditions lie in a range between the simple free stream conditions and the best guessed solution obtained from experiments, empirical relations, approximate theories, or previous computational results. For a steady flow, the better the initialization of the flow field, the faster the solution converges. There are two different initialization procedures that can be used for a composite mesh. The first method is to simply initialize all the flow subdomains with free stream conditions, however this method is computationally costly. The second method is to advance the solution on each subdomain independent of all other subdomains, using a mesh sequencing procedure, in order to pass the numerical transient state. Mesh sequencing is a method of quickly developing an approximate solution at a coarser subdomain grid level, and prolonging the solution to the next finer grid level until the characteristics of the flow are resolved on the finest level. Both initialization procedures are utilized in this study.

Boundary conditions are specified explicitly for this implicit, finite volume algorithm. There are five general boundary conditions that are used in all test

cases; solid boundary, supersonic upstream, supersonic downstream, inflow/outflow, and inter-subdomain grid boundary.

At the solid boundaries, the conditions of no-slip and impermeability with zero-normal-gradient for pressure and temperature are imposed. The density at the surface is calculated by employing the state equation.

$$u = 0, v = 0, w = 0, \frac{\partial T}{\partial n} = 0, \frac{\partial P}{\partial n} = 0 \quad (3.12)$$

Upstream boundary conditions are dependent on the flow characteristics. Supersonic inflow (excluding the boundary layer) have flow characteristics pointing from the outside toward the inside of the computational domain. Hence, the upstream boundary conditions can be specified by the supersonic free stream conditions. For the case where the upstream boundary is in the proximity of a surface, the boundary layer profile generated from the boundary layer equations is used.

The supersonic downstream flow has characteristic signals propagating from inside the computational domain to outside. Hence, the downstream boundary conditions are determined from zeroth-order extrapolation of interior variables,

$$\frac{\partial u}{\partial \xi} = 0, \frac{\partial v}{\partial \xi} = 0, \frac{\partial w}{\partial \xi} = 0, \frac{\partial T}{\partial \xi} = 0, \frac{\partial P}{\partial \xi} = 0 \quad (3.13)$$

where  $\xi$  indicates the streamwise coordinate.

Locally one-dimensional characteristic boundary conditions are used for the farfield boundaries. For each farfield cell, the normal velocity to the boundary and the speed of sound are calculated from the two-locally one-dimensional Riemann invariants given by

$$R^{\pm} = \bar{u} \pm \frac{2a}{\gamma - 1} \quad (3.14)$$

The invariants are constant along the characteristic defined by

$$\left( \frac{d\bar{x}}{dt} \right)^{\pm} = u \pm a \quad (3.15)$$

The appropriate boundary conditions are determined after the direction and magnitude of local Mach number at each cell is checked. For subsonic conditions at the boundary,  $R^{-}$  can be evaluated from free stream conditions outside the computational domain, and  $R^{+}$  is evaluated locally from the interior of the domain. The local normal velocity and speed of sound on the boundary using Riemann invariants, are

$$\bar{u}_b = \frac{1}{2} (R^{+} + R^{-}) \quad (3.16a)$$

$$a_b = \frac{\gamma - 1}{4} (R^{+} - R^{-}) \quad (3.16b)$$

The Cartesian velocities are determined on the outer boundary by decomposing the normal and tangential velocity vectors into components.

For supersonic inflow/outflow conditions at the farfield boundaries, simple zeroth-order extrapolations are used with the direction of the extrapolation dependent on the sign of the local speed of sound.

The inter-subdomain boundaries of the composite mesh, which do not coincide with the global computational domain boundaries, are required to be updated through interpolations. Because the Roe flux-difference-splitting scheme is an exact solution to an approximate Riemann problem, it is redundant to check inflow/outflow conditions, using locally one-dimensional characteristic boundary check for the boundary cells. The jump in the solution at the cell boundary is propagated in the locally correct direction and added to the existing value to get the solution at the next iteration. However,

the inflow/outflow check is necessary for a flux-vector-split [31,32] or central-differenced schemes [32,33].



## Chapter 4

### GRID OVERLAPPING METHOD

#### 4.1 Grid Generation

The current grid overlapping method allows the subdomain grids to be generated independently. Hence, a subdomain grid topology depends upon neighboring topology only to the extent, that they must overlap and that the cell sizes in the overlap region are comparable. The reasons are explained in Section 4.3. Two types of grid generation methods are used in this study, namely, the algebraic method and the Poisson's equations method.

The algebraic method is one, in which there is a known explicit functional relationship between the computational and the physical domain [34]. Hence, algebraic methods are used for simple configurations. The technique uses stretching functions to distribute points along simple analytic coordinate curves. They are effective in the area of mesh control at boundaries, but are less effective in the quality of the interior mesh points, particularly for complex domains [35]. An interactive computer program, developed by Smith et al. [36], TBGG, and based on a two dimensional algebraic two boundary grid generation technique is used in creating several subdomain grids. The essence of a two boundary method is to connect a distribution of points between inner and outer boundaries, based on a hermite cubic interpolation procedure.

For more complicated configurations an elliptic partial differential equation (PDE) approach, developed by Steger and Sorenson [37], is used to generate grids. In particular, a computer program called GRAPE developed by Sorenson [39] is used. The GRAPE program generates two-dimensional grids about airfoils and other shapes by solving the Poisson's equation,

$$\xi_{xx} + \xi_{yy} = P \quad (4.1a)$$

$$\eta_{xx} + \eta_{yy} = Q \quad (4.1b)$$

Particular parameters, such as control of the spacing between mesh points and control of the angles with which mesh lines intersect the boundaries, are incorporated into the right hand side functions P and Q. An iterative procedure is used to solve these equations.

Both codes, GRAPE and TBGG, generate two dimensional grids. Three dimensional grids are developed by simply stacking the two dimensional planes in the third dimension. Further enhancement of cell clustering within high viscous regions are accomplished by a parametric curve fitting procedure. Also, farfield rectangular subdomain grids are created using simple algebraic methods with exponential clustering in viscous regions.

#### 4.2 Overlapping Algorithm

The grid overlapping "chimera" algorithm developed by Benck et al. [13-15] is modified to serve for a multigrid, finite volume (as well as finite difference) upwind solution algorithm. The modified version is given the name MaGGiE, short for Multi-Geometry Grid Embenner. The algorithm with its modifications for a finite volume and multigrid solver is discussed initially, and then the topic of subdomain grid communications through interpolation procedures is discussed. These modifications and implementations are the bases of this study.

The program MaGGiE creates a three dimensional composite mesh from individual subdomain grids, and the necessary intergrid communications. The

subdomain grids create holes in other subdomain grids in which they are embedded or overlapped. The holes that are created in the grids are excluded from the solution. To obtain a logical sequence of grid communications between overlapped grids, a form of grid hierarchy is needed. An order of hierarchial form between the grids allows the interaction of appropriate grids, simplifies the development of the data structure required for this interaction, and limits the search to locate points in other grids for the purpose of interpolation. Grids which are on level  $L$  of hierarchy are designated  $G_{l,i}$  where 'i' is the grid index on level  $L$ . In general, grids on a given level  $L$  are partially or completely embedded in grids of level  $L-1$ . Grids on level  $L$  may overlap other grids of level  $L$ , and they may contain grids of level  $L+1$  partially or completely embedded in them. Fig. 4.1 shows an example of such a hierarchial grid arrangement.

MaGGiE's composite mesh generation consist of : (1) establishing the proper lines of communication among the grids through appropriate data structure; (2) constructing holes within grids; (3) identifying points with holes and illegal zones (solid surfaces); (4) locating points from which outer and hole boundary values can be interpolated; and (5) evaluating interpolation parameters. The MaGGiE code is divided into six stages. The first three stages are used to acquire finest level grid communication data, and the last three stages are used to acquire multigrid level communication data. Each stage is described in the following subsections, and an overview flow chart is given in Fig. 4.2.

#### 4.2.1 Hole Boundary

The composite grid generation starts with the subdomain grids being translated and rotated to their proper locations relative to fixed, global origin. If cell center interpolation data between connected grids are needed, the subdomain grids are transformed from cell vertices to cell center points. The transformed grids are created in Stage 1 and are used throughout the six stages. The cell center grids are created by averaging the coordinates of the eight cell vertices (Fig. 4.3). For example, the x-coordinate of the cell center is calculated as

$$X_{ijk}^{cell} = \left\{ X_{ijk} + X_{i+1,j,k} + X_{i+1,j+1,k} + X_{ij+1,k} + X_{ij+1,k+1} + X_{ij,k+1} + X_{i+1,j,k+1} + X_{i+1,j+1,k+1} \right\} / 8 \quad (4.2)$$

Collapsed cell centers and edge points are defined on the last grid planes in the three coordinate directions. This is done to create the same number of cell centers as there are nodes. The collapsed cell centers are calculated by averaging the four vertices of a cell surface. For example, the collapsed cell center on the KMAX plane is calculated as

$$X_{ij,kmax}^{cell} = \left\{ X_{ij,kmax} + X_{i+1,j,kmax} + X_{i+1,j+1,kmax} + X_{ij+1,kmax} \right\} / 4 \quad (4.3)$$

The edge points are defined on the IMAX, JMAX and KMAX grid corners. For example, the grid edge formed by the intersection of the JMAX and KMAX planes of the grid is calculated as

$$X_{ijmax,kmax}^{edge} = \left\{ X_{ijmax,kmax} + X_{i+1,jmax,kmax} \right\} / 2 \quad (4.4)$$

After creating the cell center grids, it is important to note that the physical space constructed by the cell centers is less than the space constructed by the nodes. Thus, care is needed in connecting cells located at and near boundaries.

A search method is used to locate the holes created in each subdomain or global grid caused by other overlapped subdomains. The search procedure can be divided into six steps.

Step 1: An initial hole boundary is specified as a surface,  $C$ , in the overlapped grid  $G_{l+1,i}$  (Fig. 4.4). The 'i' index of  $G_{l+1,i}$  will be dropped from now on for convenience.

Step 2: Outward normal vector,  $N$ , is constructed at each hole surface cell center using a vector cross product technique. Further details of this technique are given in Appendix A.

Step 3: A temporary origin,  $P_0$ , of the initial hole is located by averaging the hole surface coordinates.

Step 4: A maximum search radius,  $R_{MAX}$ , is defined as the maximum distance from the origin of the hole to a cell on the hole boundary surface (Fig. 4.5).

Step 5: The initial search determines whether a cell point ( $P$ ) from the grid  $G_l$  lies within the search radius  $R_{MAX}$ . If the cell  $P$  lies within the search circle then a vector dot product test is used.

Step 6: A vector dot product ( $N \bullet R_p$ ) is computed, where  $R_p$  is the position vector from a hole surface point to a cell point  $P$  in  $G_l$  (Fig. 4.5). If  $N \bullet R_p > 0$ , the cell  $P$  lies outside the initial hole; otherwise the cell  $P$  lies inside the initial hole and thus is defined as a hole point in grid  $G_l$ .

Figure 4.6 shows a hole and its boundary in grid  $G_l$  generated by the overlapped grid  $G_{l+1}$ . A hole point is flagged for grid  $G_l$  by setting an array  $IFLAG=0$ . A cell of  $G_l$  which is not in the hole, is flagged by setting  $IFLAG=1$ . The

next task is to locate the  $G_l$  cells which are immediate neighbors of the hole cells. These are called fringe cells, and the intergrid communication of conserved variables from  $G_{l+1}$  grid is performed on these cells. A fringe cell is also flagged IFLAG=0. The fringe and hole cells in grid  $G_l$  are shown in Fig. 4.7. A cell in  $G_{l+1}$  with the shortest distance to a fringe cell in  $G_l$  is located and called a TARGET cell. The TARGET cell is the starting point in the search for the cells are used for interpolation. The number of cells in  $G_{l+1}$ , surrounding the fringe cell in  $G_l$ , that need to be connected depends upon the order and accuracy of the interpolation procedure.

A trilinear interpolation procedure is used in the intergrid communication of conserved variables. The significance, accuracy and conservative nature of using trilinear interpolation is discussed in a following section. Once a target cell of  $G_{l+1}$  is located, a search is conducted to locate seven other cells in  $G_{l+1}$  near the target cell. The objective is to form a hexahedron which has the seven cell centers and the target cell as the vertices, such that the hexahedron includes the fringe cell of  $G_l$ . The information is transferred from the eight cells, that define the vertices of the interpolation cell of  $G_{l+1}$ , to the fringe cell of  $G_l$  using trilinear interpolation. A typical interpolation cell of a body fitted grid is a warped hexahedron. The trilinear interpolation can only be used on cubes. Each interpolation cell containing a fringe cell at which a function value is to be interpolated is mapped to a unit cube using isoparametric mapping. Isoparametric mapping [39-41] is the process of defining the same function that describes the geometry of the element as the function used to interpolate spatial variations of a variable at location P within the element (Fig. 4.8). The isoparametric mapping assumes that the transformation

between the natural  $\xi, \eta, \zeta$  coordinates and the global  $X, Y, Z$  coordinates is unique. The order of the polynomial function used to represent the field variable within an element depends upon the number of nodal variables to evaluate the coefficients of the polynomial. Hence, the interpolation cell has eight nodal variables, and thus leads to the transformation/interpolation equation of the following form,

$$f = a_1 + a_2\xi + a_3\eta + a_4\zeta + a_5\xi\eta + a_6\xi\zeta + a_7\eta\zeta + a_8\xi\eta\zeta \quad (4.5)$$

where  $a_i, i=1, \dots, 8$  are coefficients depending on the values of  $f_i$  at the vertices of the unit cube (Fig. 4.8).  $\xi, \eta, \zeta$  are coordinates of the interpolated cell,  $P$ , relative to the target cell in the unit cube. The unit cube is mapped so that  $0 \leq \xi, \eta, \zeta \leq 1$ . For example,  $a_1 = f_1$  is obtained when  $(\xi, \eta, \zeta) = 0, 0, 0$ . The other coefficients are

$$\begin{aligned} a_2 &= -f_1 + f_2 \\ a_3 &= -f_1 + f_4 \\ a_4 &= -f_1 + f_5 \\ a_5 &= f_1 - f_2 + f_3 - f_4 \\ a_6 &= f_1 - f_2 - f_5 + f_6 \\ a_7 &= f_1 - f_4 - f_5 + f_8 \\ a_8 &= -f_1 + f_2 - f_3 + f_4 + f_5 - f_6 + f_7 - f_8 \end{aligned} \quad (4.6)$$

Identifying the origin of the cube in the interpolation space relative to the coordinates in the physical space as  $(0, 0, 0) = (X, Y, Z)_{i,j,k}$ , the  $f_i$  values with the vertices become

$$\begin{aligned} f_1 &= f_{i,j,k} & f_5 &= f_{i,j,k+1} \\ f_2 &= f_{i+1,j,k} & f_6 &= f_{i+1,j,k+1} \\ f_3 &= f_{i+1,j+1,k} & f_7 &= f_{i+1,j+1,k+1} \\ f_4 &= f_{i,j+1,k} & f_8 &= f_{i,j+1,k+1} \end{aligned} \quad (4.7)$$

Note, the interpolation stencil can be identified by the target cell  $(i, j, k)$  because the other seven vertices are an extension of it. This simplifies the

storage requirements for the interpolation data, since only the information for the target cell is needed. The last agenda of the interpolation procedure is to determine the values of  $\xi$ ,  $\eta$ ,  $\zeta$  from the isoparametric mapping. The transformation data is the same as the interpolation data. The isoparametric equations mapping the interpolation space to the physical space is given by the following,

$$\begin{aligned} X &= a_1 + a_2 \xi + a_3 \eta + a_4 \zeta + a_5 \xi \eta + a_6 \xi \zeta + a_7 \eta \zeta + a_8 \xi \eta \zeta \\ Y &= b_1 + b_2 \xi + b_3 \eta + b_4 \zeta + b_5 \xi \eta + b_6 \xi \zeta + b_7 \eta \zeta + b_8 \xi \eta \zeta \\ Z &= c_1 + c_2 \xi + c_3 \eta + c_4 \zeta + c_5 \xi \eta + c_6 \xi \zeta + c_7 \eta \zeta + c_8 \xi \eta \zeta \end{aligned} \quad (4.8)$$

Note, the equations for  $X$ ,  $Y$ ,  $Z$  are the same as the Eq. (4.5), where  $f$  is replaced by  $X$ ,  $Y$ ,  $Z$ . The coefficients  $a_i$ ,  $b_i$ , and  $c_i$  are evaluated using the physical coordinates of the eight vertices of the interpolation cube. The coordinates  $X$ ,  $Y$ ,  $Z$  are the coordinates of the fringe cell,  $P$ , in grid  $G_1$ . Since  $X$ ,  $Y$ ,  $Z$  of the fringe cell are known and the coefficients are known, the interpolation data  $\xi$ ,  $\eta$ ,  $\zeta$  are found using an inverse mapping. The values for  $\xi$ ,  $\eta$ ,  $\zeta$  corresponding to the fringe cell are determined iteratively by applying the Newton's method of locating roots of a set of algebraic equations. The system of algebraic equations (Eq. 4.8) can be written in the form

$$\vec{X} = \vec{G}(\xi, \eta, \zeta) = \vec{G}(\vec{\xi}) \quad (4.9a)$$

$$\vec{R} = \vec{G}(\vec{\xi}) - \vec{X} = 0 \quad (4.9b)$$

Newton's method gives

$$\vec{\xi}^{n+1} = \vec{\xi}^n - [M_{ij}^n]^{-1} \vec{F}(\vec{X}, \vec{\xi})^n \quad (4.10)$$

for each iteration, where

$$M_{ij} = \frac{\partial F_i}{\partial \xi_j} \quad (4.11)$$



The Jacobian matrices  $M$  and  $M^{-1}$  are given in Appendix B. For each fringe cell in grid  $G_l$  the respective target cell in  $G_{l+1}$  and its interpolation data are stored.

In certain cases, the trilinear interpolation procedure fails for particular boundary cells. For those boundary cells that lie within another mesh that cannot obtain interpolation data corresponding to the three coordinate directions with values between 0 and 1, zeroth-order interpolation is used. The zeroth-order interpolation is performed from the TARGET cell, which is at a minimum distance away. There are several possible causes for failure of the trilinear interpolation or isoparametric mapping procedures. Failure can occur if the interpolation cell, that contains the boundary cell is extremely warped, which may cause improper transformation to the cube space, or if the Newton's iterative method of determining  $\xi$ ,  $\eta$ ,  $\zeta$  from the system of equations fail. A loss of accuracy occurs at these cells with zeroth-order interpolation. Because the number of zeroth-order cells is usually less than five percent of the total number of boundary cells, this method is usually acceptable. Only two of the six test cases (Chap. 5) contained boundary cells which use zeroth-order interpolations. The inclusion of zeroth-order interpolation procedure in MaGGiE increases the robustness of the grid connection algorithm for subdomains of different topologies.

#### 4.2.2 Outer Boundary

The procedure described in Stage 1 for fringe cells is repeated in an opposite manner at the outer boundary cells of the overlap region where information is transferred from the grid  $G_l$  to the grid  $G_{l+1}$ . Again, target cells in  $G_l$  and interpolation data are determined for the outer boundary cells in grid  $G_{l+1}$ .

### 4.2.3 Output Format

In Stage 3, illegal communications between subdomain grids are checked, and grid connections with interpolation data, for the finest grid level, is written in a vectorized form as an output. Illegal communication between subdomains occurs, when one or more of the interpolation cell vertices is a fringe cell or an outer boundary cell (Fig. 4.9). In Fig. 4.9, an interpolation cell of grid  $G_i$  includes a fringe cell as one of its eight vertices. Information is being transferred from the eight vertices to an outer boundary cell in  $G_{i+1}$ . Simultaneously, the fringe cell in  $G_i$  is receiving its information from an interpolation cell in  $G_{i+1}$ . Hence, there is a redundancy of information being passed between the grids  $G_i$  and  $G_{i+1}$ , and most importantly, this causes a loss of conservation across the boundaries. The risk of illegal communication between fringe cells of  $G_i$  and outer boundary cells of  $G_{i+1}$  is decreased with the increase in the width of the overlapped region, and the reduction of the order of accuracy in the interpolation procedure (or reduction in the width of the interpolation stencil). However, reducing the order of accuracy of the interpolation reduces the accuracy of the global solution on a subdomain.

For each mesh the following information is given:

- (1) vector sets (  $JI(i)$ ,  $KI(i)$ ,  $LI(i)$  ), which contain the indices of the reference cell for each interpolation stencil,
- (2) the corresponding interpolation coefficients (  $DXI(i)$ ,  $DYI(i)$ ,  $DZI(i)$  ),
- (3) vector sets (  $JB(i)$ ,  $KB(i)$ ,  $LB(i)$  ), which contain the indices of cells in mesh  $G_i$ ,  $G_{i+1}$ , etc. that have values interpolated from other grids.
- (4) a cross-index list,  $IBC$ , which is a pointer to the updated boundary values that are retained in memory in a single-index list,  $QB$ , of the flow solver.
- (5) the  $IFLAG$  array, which defines holes cells by the value  $IFLAG = 0$

A brief description about how the output data is used by the flow solver is summarized in section 4.7

#### 4.2.4 Composite Grids for Multigrid Method

The objective of Stages 4,5 and 6 is to obtain grid connection data for coarser grid levels, so that the information can be used by a multigrid flow solver for composite meshes. The coarser level grids, say  $M$ , are generated from the finest level grids of each subdomain as explained in Section 4.4. One of the criteria used in creating coarser level composite grids, is to create the holes in such a manner, that during a restriction stage of a multigrid cycle, the restricted functional values are not contaminated by the hole cells on the finer level grids. Secondly, the hole cells of the coarser grids,  $G_l$ , are connected to the cells at the coarser levels of other grids,  $G_{l+1}$ . This is done to avoid the contaminated information being transferred from within a hole of the coarser grids to non-hole cells of the next finer grids, during the prolongation stage.

The hole cells in the coarser grids are created from holes in the finest level grids of the composite mesh. A search sequence of locating eight finer level cells that make up a coarser level cell is accomplished, such that, if at least one of the eight finer cells is a hole cell, which is designated  $IFLAGM=0$ , then the coarser level cell is designated  $IFLAGM=0$ . If none of the eight finer cells are hole cells, then the coarser level cell is an exterior cell, and it is designated  $IFLAGM=1$ .  $M$  denotes the coarseness level of the grid. The above procedure of defining holes in the coarser subdomains eliminates the restriction errors caused by the holes in the finest level mesh. There are no restriction errors because the restricted value of a coarser cell is determined by weighted values of eight finer cells that make up the coarser cell. The above sequence is repeated for each coarser level of the composite mesh.

Once the hole cells in the coarser subdomain grid,  $G_l$ , are located, a search is conducted for interpolation data for these cells, with  $IFLAGM=0$ , from a coarser subdomain grid,  $G_{l+1}$ . This search can only fail for those cells of  $G_l$  at level  $M$  which coincide with, for example, the body around which the grid  $G_{l+1}$  is generated. Such a zone is designated the ILLEGAL ZONE (see Fig. 4.10), and their cells are left with the flag  $IFLAGM=0$ , as they are effectively excluded from this coarse level flow calculations. The  $IFLAGM$  values of all the other cells, which now have interpolation data, are switched from 0 to 1, and are included within the calculations on the coarser subdomains. They are switched, because these cells are used to prolonge their functional values to the next finer level excluding the illegal zone.

Stage 5 of MaGGiE locates the outer boundary connection cells of the overlapped region, where the interpolation is accomplished from the coarser level of  $G_l$  to the coarser level  $G_{l+1}$ . Such outer boundary cells of  $G_{l+1}$ , for which interpolation data are now available, are flagged as  $IFLAGM=0$ . Note, that on the coarser grid levels, the definition of the overlapped region between grids is changed. It is no longer an outer-band region around the embedded grid. Instead, the overlapped region becomes the entire hole region defined by the finest level embedded grid. The results of this change allows proper prolongation to occur in the multigrid flow solution algorithm. All of the information obtained above for the hole cells, illegal zones, and outer boundary cells is written in a data vector form for the multigrid solver in Stage 6 of MaGGiE.

An option is built into this algorithm, where one can choose the grid level of  $G_{l+1}$ , from which the interpolation is to be performed to the coarse level of  $G_l$ . The obvious choice is searching interpolation data between grids on the same level. If the cells involved in the interpolation are of comparable sizes at

the finest level ( as it is desirable for accuracy), they are again of comparable sizes at the coarser levels. This option increases the success in forming the interpolation cells, but it decreases the accuracy because interpolated values are not properly averaged over the entire physical space that the coarser cell occupies. Also, this option eliminates the possibility of mesh sequencing [21], where Eq. (2.1) is solved at the same coarse level of all the subdomains until some convergence is reached.

#### **4.3 Overlapped Region in a Composite Mesh**

The width of the overlapped region is dependent on the width of the interpolation formula, the stencil of the spatial differencing, and the smoothness of cells. Too much overlap between subdomains results in unnecessary duplication of computations in these regions and too little overlap results in illegal or lack of communications between subdomains. Five to ten cells overlap is found to be efficient for finest level grids. The objective is to create each subdomain grid independently in such a manner that when one grid is overlapped/embedded within another the cell sizes of both grids are of the same order within the overlapped region. This is not a necessary condition, however the transference of solution from one grid to another through interpolation becomes more accurate the closer the cell sizes are. The accuracy improves for similar cell sizes because most interpolations functions are weighted by physical distances and not percentage of cell volumes.

#### **4.4 Inter-Subdomain Conservation**

For subdomain grids, which in general overlap each other in an irregular fashion, it is desirable to use conservative interface procedures. Such a practice helps, for example, finding the correct shock location for shocks passing through grid boundaries, and ensures artificial shocks are not generated at grid interfaces. This section introduces some of the approaches

currently being considered to maintain conservation at the overlapped boundaries. It should be noted that this study uses the nonconservative trilinear interpolation approach for intergrid communication. The trilinear interpolation has been discussed in Section 4.2.1.

A preliminary study was done by Berger [42] on a general procedure, for deriving conservative interface conditions that give weak solutions to the differential equation, if they converge on one and two dimensional overlapped grids. Let  $U$  be a weak solution to a hyperbolic system of one dimensional conservation laws,

$$U_t + f(U)_x = 0 \quad (4.12a)$$

$$U(x, t=0) = U_0(x) \quad (4.12b)$$

which satisfies the integral equation

$$\int \int U \Phi_t + f(U) \Phi_x dx dt + \int U_0(x) \Phi(x) dx = 0 \quad (4.13)$$

for any smooth test function  $\Phi(x,t)$ . The conservative interface conditions can be derived based on the direct numerical approximation of Eq. (4.13). If a conservative scheme is multiplied by  $\Phi(x,t)$ ,  $\Delta x$ , and  $\Delta t$  and summed over all grid points, it can shown that a discrete approximation to the integral converges exactly by the numerical scheme. A discrete approximation of the integral,

$$S = \int u(x, t) dx \quad (4.14)$$

is

$$S = \sum_j h u_j + O(h^2) \quad (4.15)$$

using the trapezoidal rule. The approximation is generalized at the boundaries of irregular grids, such as, the overlapped grids. Alternatively, the

conservative interface condition can be derived using a finite volume approach, and balancing the spatial flux at the interface (in two dimensions),

$$\int \int F^{(1)} d\xi d\eta = \int \int F^{(2)} d\xi d\eta \quad (4.16)$$

where  $F^{(1)}$  and  $F^{(2)}$  are the spatial fluxes of grid (1) and grid (2) in the overlapped region, respectively. In the general two dimensional case, the interface equation for the flux across the interface cell is a linear combination of the neighboring fluxes,

$$F = \sum_i^N \alpha_i f_i \quad (4.17)$$

with coefficients,  $\alpha_i$ , determined by the amount of overlap and the integration rule. The steps to implement such a flux balancing is to determine the weight of the cells in the integration rule, and determine the amount of the main grid's flux to be apportioned to the boundary cell. Because two quadrilateral grids can intersect in a many sided polygon, depending on the mesh ratios of the grids, these steps can be complicated, and when extended to three dimensional grids they can become too expensive and complicated to warrant their usefulness.

An alternative to conserving the spatial flux across overlapped boundaries is to conserve the time flux,  $Q$ , of the cell center at the boundaries. The conserved variables  $Q$  refer to the time flux of mass, momentum, and energy. In the overlapped region, the conserved time flux can be expressed in three dimensions as

$$\int \int \int Q^{(1)} d\xi d\eta d\zeta = \int \int \int Q^{(2)} d\xi d\eta d\zeta \quad (4.18)$$

where  $Q^{(1)}$  and  $Q^{(2)}$  are the time fluxes of grid (1) and grid (2) respectively. The time-flux conservation approach has been found to maintain the

conservative properties at the boundaries within truncation errors [43,44]. The conservation of time flux is accomplished by interpolation to the cell centers of one grid assuming a weighted variation of time flux with the cells of the other grids. For the interpolation procedure to be conservative, the weighted variation of the time flux should be dependent on the percentage of volume of the cells of  $G_l$ , that overlap a cell in  $G_{l+1}$ . However, to find the cell volume weighted variations for three dimensional grids is a geometrically complicated procedure that usually cannot be generalized.

Another approach, which is used in this work, is to use a nonconservative interpolation procedure that has the same properties as the conservative procedure. In the nonconservative approach, the weighted variations are usually dependent on the linear distances between a boundary cell and its surrounding interpolation cells. Nonconservative interpolation assumes continuity of the interpolant. Polynomial expansions can be used as interpolation functions. The number of coefficients in the polynomial should equal the number of nodal variables available to evaluate these coefficients. Linear variation of a variable within an element can be expressed by functional values at the nodes. For example, a hexahedron element in three dimensional grid has eight nodes or vertices; hence, an incomplete polynomial expansion of eight terms in the linear form (Eq. 4.5) is used in the interpolation procedure. This interpolation procedure is also known as trilinear interpolation which was discussed previously. Higher order interpolation methods, such as, quadratic and cubic variations, require additional installation of nodes at various points within the element. However, interior nodes are undesirable because additional nodes lead to complications in formulation and computations.



#### 4.5 Global Accuracy

An important issue in regards to the interpolation is its effect on the overall accuracy of the solution. The degree of continuity and the amount of conservation of the flow variables that can be maintained have been investigated previously [19,20]. For the overall accuracy to be as good as the discretization formula used for Eq. (2.1), it is shown that the width of the interpolation formula should be  $(1/4 pr + 1)$  if the width of the overlapped region is constant. In this formula,  $p$  denotes the order of the differential equation being solved, and  $r$  denotes the order of accuracy of spatial discretization. Hence, if the differential equation is of order two and spatial discretization is of order two, then the width of the interpolation formula is two. Two sets of fringe and outer cells are needed for second-order accurate matching of the solution to second-order differential equations being solved here. Having two sets of boundary cells, in effect, is transferring fluxes from grid  $G_{l+1}$  to grid  $G_l$ . However, using a second set of boundary cells increases the width of the needed overlapped region to ensure correct grid connections without illegal communications. The risk of illegal communication between fringe cells of  $G_l$  and outer cells of  $G_{l+1}$  is increased by either having two sets of boundary cells, or a smaller overlapped region. Also, increasing the width of the interpolation formula increases the storage memory and the run time of the flow solver, because there are twice as many cells to update.

#### 4.6 Modified Solution Algorithm

Modifications are made to the implicit, finite volume, multigrid algorithm (Chap. 3), in order to recognize the multiple, overlapped subdomains with holes. The standard block or scalar tridiagonal inversions for Eq. (2.1), or its approximate diagonalized counterpart, are altered for the case of overlapped grids with holes. To treat the hole, fringe, and outer boundary cells, the

following modifications are made: (1) set the off-diagonal 5x5 blocks in the coefficient matrix to zero, (2) set the diagonal elements of the diagonal block to unity, (3) set the residuals to zero. This results in the computed values of  $\Delta Q$  for these cells to be zero. Since  $\Delta Q$  of the fringe and outer boundary cells are zero, the specified boundary conditions for these cells are automatically preserved. For example, let one of the spatial factors of Eq. (2.1) be written as a system of algebraic equations in block tridiagonal form as

$$a_{ij} * \Phi_i = R_i \quad (4.19)$$

where  $a_{ij}$  are the 5x5 blocks of coefficient tridiagonal matrix.  $\Phi_i$  are the unknown vectors, which represent  $\Delta Q$  of Eq. (3.7). The right hand side residual is represented by  $R_i$  which are the known vectors. Then,

$$R_i * IFLAG_i \rightarrow R_i \quad (4.20a)$$

$$a_{ij} * IFLAG_i \rightarrow a_{ij} \quad , \quad i \neq j \quad (4.20b)$$

$$(a_{ij} * IFLAG_i) + (1 - IFLAG_i) \rightarrow a_{ij} \quad , \quad i = j \quad (4.20c)$$

where  $(\rightarrow)$  indicates that its right hand side is to be replaced by its left hand side (Fig. 4.11). The IFLAG is zero or one, depending on if IFLAG is a hole cell or not. The diagonal elements of the block  $a_{ij}$  are set to unity. If the approximate diagonal form is used, then this process is repeated three times for each direction. The discretization of the right-hand side of Eq. (2.1) uses a five-point stencil for second-order spatial accuracy. In order to avoid the erroneous flux from a cell in a hole, when computing a cell neighboring a fringe point, the  $Q$  value of the neighboring hole cell is set to the  $Q$  value of the fringe cell.

In the current domain decomposition method the existence of holes caused by embedded or overlapped grids complicates the implementation of a standard multigrid algorithm. If the standard multigrid algorithm is used for multiple subdomains with holes, the restriction and prolongation stages would use cells

which lie within holes, to transfer information from coarser to finer grids and vice versa. This is an undesirable trait of multigriding for multiple subdomains, for the obvious reason, that the hole cells do not contain correct solution information of the subdomain.

The procedure for creating holes and illegal zones in coarser level grids from the finer level grids described previously in Section 4.2.4 eliminates the problems that occur in the restriction stage. Any cell in the coarser level grid, that obtains a volume weighted restriction value made up of at least a hole cell in the finer level grid, is labeled as a hole cell. However, these hole cells on the coarser level grids, excluding the illegal zone cells, have interpolation data obtained from the connected grids. Thus, they can be used in the restriction and prolongation stages. The solution is transferred between the coarser grids,  $G_j$  and  $G_{j+1}$ , over a larger physical domain than the overlapped regions of the finest level meshes. There is an increase in the physical domain where updating occurs. However, the actual number of cells being updated in the coarser level grids is usually less due to the decrease in the number of grid cells. Each coarser level three dimensional grid reduces its number of cells by a factor of  $1/8$  of its finer level grid cells. Also note that the standard block or scalar tridiagonal inversions for Eq. (2.1) are executed at the coarser levels, after the modifications are made analogous to Eq. (4.20) with  $IFLAG_j$  being replaced by  $IFLAGM_j$ .

Modification is needed in the prolongation stage to nullify the weight of the contributions from the illegal zone cells. Note that no such modification for the illegal zone is needed in the restriction stage, because the illegal zone of the coarser grid is inside the hole of the finer grid. The prolongation is performed from the coarser cells, say  $C1$  and  $C2$ , to pseudo finer level cells, say  $f1$  and  $f2$  ( $f1$  is closer to  $C1$  than  $f2$ ), in one direction as,

$$\Delta Q_{f1} = (A1 * \Delta Q_{C1}) + (B1 * \Delta Q_{C2}) \quad (4.21a)$$

$$\Delta Q_{f2} = (A2 * \Delta Q_{C1}) + (B2 * \Delta Q_{C2}) \quad (4.21b)$$

where

$$A1 = IFLAGM_{C1} [1 - (b * IFLAGM_{C2})] \quad (4.22a)$$

$$B1 = IFLAGM_{C2} [1 - (a * IFLAGM_{C1})] \quad (4.22b)$$

$$A2 = IFLAGM_{C1} [1 - (a * IFLAGM_{C2})] \quad (4.22c)$$

$$B2 = IFLAGM_{C2} [1 - (b * IFLAGM_{C1})] \quad (4.22d)$$

and  $a, b$  are the bilinear interpolation constants (0.75 and 0.25). The  $IFLAGM=0$  flags the illegal zone cells at the coarser level. This process is repeated in the second direction, using the pseudo finer level cells of the first direction. Finally, when this process is repeated in the third direction with Eq. (4.21), and using the finer level cells of the second direction, the corrections are recovered for the actual finer level cells. The results of this process is a trilinear interpolation with small bias around the illegal zone. More details of the modified algorithm is given by Baysal et al. [44] and Fouladi [45].

#### 4.7 Procedure for Solution Algorithm

The steps to advance a subdomain solution of a composite mesh are: (1) update the boundary cells of a subdomain cells by using specified boundary conditions or interpolation values; (2) solve the subdomain flow field with the implicit, finite-volume, upwind scheme; (3) interpolate the conserved variables for intergrid communications for the boundary cells of other subdomains; (4) repeat steps 1 to 3 for each subdomain mesh of the composite mesh in the hierarchial order. Hence, there are two functions that the flow solver must perform on the interpolation data. The first function is to update the interpolation boundaries, and the second function is to interpolate data for

the boundaries of the subdomain grids. To accomplish both of these tasks, the interpolation data structure is of vector form (refer to Section 4.2.3).

The flow solution on a subdomain  $G_l$ ,  $G_{l+1}$ , etc. is advanced in time. The interpolation boundary cells defined by the indices  $JB(i)$ ,  $KB(i)$ ,  $LB(i)$  are updated from the flow solver,  $QB$ , list of interpolated values through a cross index array  $IBC(i)$ . Next, the solution of the subdomain is advanced excluding the hole cells which are designated by  $IFLAG=0$ . From the advanced solution, the interpolation values for other connected subdomains are solved using the interpolation reference cells given by the indices  $Jl(i)$ ,  $Kl(i)$ , and  $Ll(i)$ . These interpolated values for connected grids are then loaded into the  $QB$  vector to be used in updating boundaries of overlapped subdomains. The process is repeated for each solution iteration on the composite grid. The number of solution iterations on a subdomain grid at one time is case dependent. However, it is computationally more efficient to have more than one iteration step at a time, since the process of obtaining interpolation data, updating, and switching the solution algorithm from one subdomain to another, are costly. The problem with having more than one iteration performed on a subdomain, is that the interface boundary conditions are frozen for those iterations. More than one iteration on each domain is permissible if a converged steady state solution is sought, but if the flow is unsteady or time dependent, then one iteration per grid is necessary for a time accurate solution.

## Chapter 5

### APPLICATIONS OF GRID OVERLAPPING

Six composite grid cases are considered in developing and demonstrating the Multi-Geometry Grid Embedder scheme. Composite grids and connection data (i.e. interpolation data) are obtained for all six cases, and computational flow results are shown for five of these cases [44,45]. These cases are:

- (1) Single level composite grid and connections for a blunt-nose cylinder grid (BNC) embedded within a Cartesian farfield.
- (2) Multigrid composite grids and connections for BNC embedded within a Cartesian farfield.
- (3) Composite grid and connections for BNC overlapped with an outer grid of similar topology (C-O).
- (4) Composite grid and connection for an ogive-nose cylinder (ONC) in the proximity of a flat plate.
- (5) Composite grid and connections for a cylindrical store model connected to an L-shaped sting embedded within a Cartesian farfield.
- (6) Composite grid and connections for a cylindrical store model with fins and a curved sting positioned above a rectangular cavity.

The first three cases are for the same blunt-nose cylinder (BNC) geometry, which are investigated in [44,45]. These three cases are used to validate the finite volume grid connections procedure for single and multigrid levels and to check the conservation across grid boundaries. The last three cases are to

test the versatility of MaGGiE for more complex composite mesh configurations.

Note that the composite meshes and solution contour plots given for each case are cell centers. The plots are based on the cell centers to eliminate the probable errors in representing the solution near jagged hole boundaries, caused by the normal averaging of the finite volume solution, to obtain grid nodal values. This allows correct assessment of contour lines crossing interface boundaries without error produced in the averaging procedure.

### **5.1 Case 1: Blunt-Nose Cylinder in a Cartesian Grid (Single Level)**

A boundary fitted C-O grid, which is wrapped around the BNC, is embedded completely within a Cartesian global grid (Fig. 5.1). The flow Mach number is 1.6, the Reynolds number is  $2 \times 10^6$  per foot, and the total temperature is 585 degrees Rankine. The BNC is at  $32^\circ$  angle of attack. The rationale behind this test case is threefolds. (i) A simple body-fitted grid for a body of revolution, such as a C-O grid, is topologically very different than a Cartesian grid. (ii) There is a computational solution available for this case which is obtained on a C-O grid only [21], i.e. without overlapped grids (iii) There is experimental data available for comparisons [46].

The blunt nose cylinder has a base diameter (D) of 3 inches and a length of 20 inches (Fig. 5.2). The body fitted C-O grid around the BNC is generated using program GRAPE. It is used to generate a two dimensional C-type grid around half of the BNC. The C-grid is rotated 360 degrees around the body centerline to create a three dimensional C-O grid (Fig. 5.3). Clustering in the viscous region near the body is enhanced by a simple parametric curve fitting procedure, which uses the Bose-Einstein Function,

$$F = S_1 + \frac{S_2 - S_1}{1 - e^{-k(x-x^*)}} \quad (5.1)$$

where

$$S_i = Ax + B \quad (5.2)$$

and  $x^*$  is the intersection of lines  $S_1$  and  $S_2$  in a coordinate space where  $0 < x < 1$  and  $0 < S < 1$ . (A) and (B) are the slope and starting values of the lines  $S_i$ , respectively. (F) is the parametric curve fitting function. The Cartesian outer grid is generated using a simple algebraic technique. Clustering is done in three directions, in the region where the BNC is embedded, to ensure consistent cell sizes within the overlapped regions between the two grids (Fig. 5.4). Both meshes, BNC and Cartesian, are simply generated grids, and hence provide an excellent test case for creating a composite mesh. The BNC mesh is positioned in the center of the Y-Z planes of the Cartesian mesh (Fig. 5.4). The composite grid dimensions are given in Table 5.1.

The composite mesh and its interpolation data is generated by MaGGiE. A summary of the number of hole cells and boundary cells is given in Table 5.1. The hole boundary cells in the Cartesian mesh and the outer boundary cells in the C-O grid are all connected using trilinear interpolation, i.e. none of the boundary are connected using zeroth order interpolation. Shown in Fig. 5.5 is the hole boundary surface in the Cartesian grid created by the embedded C-O grid. These hole boundary cells of the Cartesian grid are connected to the C-O grid within the overlap region. The overlap region, including outer boundary cells in the longitudinal and crossflow planes, are shown in Figs. 5.6, 5.7, respectively. The outer boundary cells of the C-O grid are connected to the Cartesian grid within the overlap area. All communications between the two subdomain meshes take place within the overlap region. A ten-cell overlap between grids is specified to ensure that no illegal communications between hole cells and boundary cells occur (Figs. 5.6, 5.7). The ten-cell overlap is not



necessary for proper grid connections; however, it is not the intent of this work to determine the optimum width of the overlap region, but instead to test the feasibility and accuracy of MaGGiE grid connection results.

Initially a mesh sequencing strategy is used on the individual subdomains to reduce the amount of computational time required to overcome the initial transient states [44,45]. The logarithm of the residual history for the finest grid level solution of the composite mesh is shown in Fig. 5.8. Convergence rate indicated in 900 work units is approximately 0.99. Work units (WU) is defined as the total CPU time divided by the CPU time for one global iteration,

$$WU = \text{CPU}_{\text{tot}} / \text{CPU}_{\text{iter.}} \quad (5.3)$$

The longitudinal pressure coefficient (  $C_p$  ) distribution on the leeside surface of BNC is shown in Fig. 5.9. Superimposed on this figure are the results from [21,44,45]. The results obtained on a single C-O grid are slightly better than those on the composite grid. This is somewhat anticipated since  $\xi$ -constant surfaces of C-O grid follow the flow stream surface closer than the  $\xi$ -constant surfaces of the Cartesian grid. Presented in Fig. 5.10 are the normalized pressure contours of the longitudinal symmetry plane for the composite mesh and the single C-O grid without embedding. It should be noted here that when using the three dimensional data of different subdomains to plot in two dimensions, one can not find longitudinal or lateral surfaces of these subdomains which match in location or in curvature. This results in some discrepancies across boundaries. Also the postprocessing of the data, especially in curve fitting near intergrid boundaries, is restricted to the capabilities of the plotting programs. In any event, the contour lines cross the intergrid boundaries rather smoothly. The shock freely crosses the interface

between subdomains with the correct angle. The contour lines of the composite mesh match quite well with that of the single C-O grid. The crossflow density contours at the base of the BNC are shown in Fig. 5.11a. The leeside vortices pass freely through the intergrid boundary. Some minor distortions may be attributed to the fact, that the Cartesian section of the crossflow surface in Fig. 5.11a is planar one, as opposed to that of the single C-O grid (Fig. 5.11b) which is curved. Their trends and magnitudes, however, agree very well. Qualitatively, there is little loss of conservation across boundaries in the streamwise as well as crossflow directions. The combination of the trilinear interpolation at intergrid boundaries and the use of Roe flux-difference-split scheme appears to maintain time conservation across intergrid boundaries.

## **5.2 Case 2: Blunt-Nose Cylinder in a Cartesian Grid (Multi-Level)**

The second case involves the C-O grid of the BNC embedded completely within the Cartesian global grid in the same manner as in Case 1. The composite mesh flow conditions are also identical to those of case 1 (Section 5.2). This case is used to test the coarser subdomain level intergrid communications with the multigrid, finite volume solution algorithm. The results are compared to the computational solution on a single C-O grid, and the experimental data, as well as the results on the composite mesh without multigrid acceleration. Two levels of intergrid boundary interpolation data are generated by MaGGiE. Two grid levels are considered minimum to verify the plausibility of using a multigrid solution scheme on a composite mesh. The first and second level grid dimensions are given in Table 5.1. Figure 5.12 shows the coarser level composite mesh.

An illegal zone of one cell from the surface of BNC on the coarser grid level is specified (Fig. 5.13). The illegal zone created in the Cartesian mesh on the second level is shown in Fig. 5.14. The number of illegal zone cells is given in

Table 5.1. All of the coarser level hole cells of the Cartesian grid and the outer boundary cells of the C-O grid are connected using trilinear interpolation, i.e. no cell on the coarser subdomain grids are connected using a zeroth order interpolation.

The solution obtained [44,45] using the multigrid scheme are the same as the single-grid scheme. Therefore, to eliminate repetition, contour plots are not given. The same initialization of subdomains used in the single-gridding is used as the starting solution in this multigridding case. This ensures accuracy in comparing the two cases. The effectiveness of the multigrid scheme on time history of the residual of the finest grid level of BNC is shown in Fig. 5.8. Convergence rate obtained in 900 work units is approximately 0.98. If more than two levels of grids were used, the savings in CPU time would be more dramatic. This would require the interpolation information to be generated at more than two levels of grids. The increase in the number of coarser levels do not greatly increase the amount of storage memory for interpolation data, because the number of hole cells needing interpolation data decreases with each coarser grid level. Most of the increase in memory is due to the modified multigridding algorithm for the composite mesh. The two-level multigrid computations of the BNC case requires 25 megawords of computer memory, as opposed to 21 megawords for single-level computations on the fine grids.

Each multigrid V-cycle consists of one time step calculation on two grid levels of each subdomain. This cycling strategy is chosen over the more computationally efficient alternatives (such as more time steps on each level or more multigrid cycles on each subdomain before switching). These alternatives seem to be more economical (computation time), but may result in inaccuracies in the solution due to frozen interpolated values of the intergrid boundaries.

Overall, the multigrid inter-subdomain communications data compares well to the single-level results, and offers a big advantage in computing (timewise) the flow on a composite mesh.

### 5.3 Case 3: Blunt-Nose Cylinder With Two Overlapped, C-O Grids

Grids which overlap at a surface, similar to grid patching, is denoted here as simple overlapping. The C-O grid of the blunt-nose cylinder is simply overlapped with an outer C-O grid of similar topology. The purpose of this case is twofolds:(1) to check the finite volume grid connection procedure for simply overlapped grids, (2) to eliminate some of the problems associated with plotting the flow solutions from multiple subdomains, in order to check the smoothness of contour lines as they pass across these interfaces. For this case the  $\eta$ - $\zeta$  surfaces of the two C-O grids lie in the same plane.

Both the outer and inner C-O grids are created from the original single C-O grid of dimensions  $81 \times 65 \times 57$  with respect to  $\xi$ ,  $\eta$ ,  $\zeta$  coordinates. The inner BNC C-O grid is created by discarding the last 16  $\zeta$ -planes of the original C-O grid (Fig. 5.16), and the outer shell is created by removing the first 32  $\zeta$ -planes (Fig. 5.17). The outer grid is also rotated two degrees, so that the  $\eta$  constant lines do not match between the two grids. Hence, when the two subdomain grids are combined to create a composite mesh, an overlapped region is formed with  $\xi$  and  $\zeta$  constant lines being identical. The  $\eta$ -constant lines within the overlapped region do not match, because of the outer grid rotation of two degrees.

A problem arises when the general grid overlapping/embedding procedure is used for grids overlapping along a constant boundary surface. In the general procedure, a hole is generated in a grid by another grid. Communication between grids occur within the overlapped region. Simply overlapped grids do not generate holes. There are two different overlapping

methods that can be used in this case. The first method is to simply connect the overlapped grids at their outer boundary cells, hence there are no fringe cells, or hole cells generated. The second method, which is used in this case, is to define the interface boundary surface as a hole. This creates a hole boundary neighboring the hole in the connected mesh. For this case, the hole is specified, such that the hole cells are on the  $\zeta=1$  constant surface and the hole boundary is at the  $\zeta=2$  constant surface of the outer C-O grid. The outer boundary of the inner C-O grid is specified at  $\zeta=41$  surface. An eight cell overlapped region is defined between the connected inner and outer C-O grids.

Shown in Fig. 5.18 is the hole boundary surface in the outer C-O grid and the outer boundary surface in the inner grid, where the interpolation is being performed. The overlapped region in the longitudinal and crossflow directions are shown in Fig. 5.19 and Fig. 5.20. The grid dimensions and the number of hole cells and boundary cells are given in Table 5.1.

The normalized pressure contours on the longitudinal symmetry plane of the composite mesh are presented in Fig. 5.21 [44,45] The contour lines compare well with that of the single C-O grid without embedding. The interface boundary between the two grids is shown by a border line, separating the two sets of contours. Although there are two degrees difference in the longitudinal  $\eta$ -planes between the outer and inner grids, the contour lines cross the interface in a continuous manner with no jumps unlike what is shown in Fig. 5.10 of Case 1. The longitudinal planes of these grids are closer in the physical space than those of Case 1 involving the C-O and Cartesian grids. Because of this, the contour lines are more continuous and represent closer to the actual values across the interface. Hence, most of the jumps in the contour lines across the interface in Fig. 5.10, may be contributed to the inadequacies in plotting procedures and not the interpolation method.

The normalized crossflow density contours are presented in Fig. 5.22. The contours at the base of the blunt-nose cylinder lie on the same  $\xi$ -plane for the two grids. Hence, assessment of the contour lines crossing the interface can be made without plotting errors. Again, the contours compare well with respect to the solution on the single C-O grid shown in Fig. 5.10. The lines are continuous across the hole boundary. However, in the outer grid near the interface, the density contours slightly change angles with respect to the inner grid lines. They correct themselves away from the boundary. This phenomena is not seen in the composite grid and the single C-O grid shown in Fig. 5.11. The cause could be related to a slight loss of conservation across the hole boundary.

#### 5.4 Case 4: Ogive-Nose Cylinder Near a Flat Plate

The fourth case is the flow past an ogive-nose cylinder (ONC) with a sting in the proximity of a flat plate wing. The flow is turbulent at zero angle of attack with Mach number 2.86, Reynolds number  $2 \times 10^6$  per foot, and the total temperature of 610 degrees Rankine. Because there are two components of different geometries in this configuration, employing simple grids requires the overlapping technique. The objective of this case is to apply the overlapping method to a more complicated flow interaction between bodies of nonsimilar topologies. There are alternative methods to discretize the domain of a cylinder near a flat plate without using the present method. These methods [47,48], however, can not use simple grids such as a C-O grid and a Cartesian grid. They require three dimensional surface grid generators. Furthermore, the grid interfaces need to be planar or the grid lines going across these interfaces need to be continuous.

The C-O grid of the ONC is embedded in the Cartesian grid of the flat plate wing at a nondimensional distance of  $Z/D=3.5$ , where  $Z$  is the normal coordinate direction and  $D$  is the diameter of the cylinder. The base diameter

and the length of the cylinder are 1.2 in. and 21.6 in., respectively (Fig. 5.23). Shown in Figs. 5.24, and 5.25 is the composite mesh. The ONC grid is created in a similar fashion to the BNC grid of Case 1. A two dimensional C-type grid is generated using the program GRAPE, and rotated 360 degrees about the centerline axis to generate the three dimensional C-O grid (Fig. 5.26). The Cartesian grid is generated with exponential clustering in the viscous region near the flat plate wing. The grid dimensions and physical domain are given in Table 5.1.

The composite mesh is created with a 7 cell overlapped region between the Cartesian and the C-O grids. Shown in Fig. 5.27 and 5.28 is the composite mesh with the overlapped region. The hole boundary is created in the Cartesian grid, such that in the  $\zeta$ -direction, 10 cells of the ONC grid lies within the hole. Because the ONC lies near the flat plate, the outer boundary of the C-O grid needs to be within the physical boundary of the Cartesian grid for proper ONC outer boundary connections. Hence, the distance of the ONC above the flat plate is a grid constraint for the ONC grid. If the ONC grid is allowed to extend out of the Cartesian grid, below the flat plate, an irregular boundary surface is needed, The MaGGiE code is capable of handling such a case, if the flat plate is considered a boundary that creates a hole in the ONC grid. The part of the C-O grid lying outside of the Cartesian grid is considered to be the hole. The hole boundary created in the C-O grid lies within the Cartesian grid. Hence, an irregular boundary surface is created for proper connection between the two grids. However, this case is not done to minimize the complexities of creating a composite mesh. The number of boundary and hole cells of the composite mesh are given in Table 5.1.

The flow solutions presented in Figs. 5.29-32 for this case lacks experimental data for comparison [44,45]. The Mach number contours of the

longitudinal plane of symmetry are shown in Fig. 5.29. An enlarged view near the grids' interface is given in Fig. 5.30. The contours cross the boundary somewhat smoothly with small jumps due to mismatch of grid planes in the plotting procedure. The interaction of the cylinder forebody shock, and the boundary layer of the flat plate, is followed by the reflected recompression waves impinging on the cylinder aftbody. The influence of the reduced pressures in the region between the cylinder and the flat plate is observed as a reduction of  $C_p$  values on the flat plate (Fig. 5.31). They are slightly negative almost everywhere except in the region where the shock impinges. The interference of the flows is further demonstrated by the Mach number contours at the cross flow plane at the base of the nose (Fig. 5.32). The shock imparts a significant momentum on the fluid particles in the normal and spanwise directions.

#### 5.5 Case 5: Store Model with L-Sting in a Cartesian Grid

Case 5 involves a more complicated composite mesh made up of three different grids, each with a different topology. The composite mesh is created for an ogive-nose store connected to a L-shaped sting in a Cartesian farfield (Fig. 5.33). The flow Mach number is 1.6, Reynolds number is  $2 \times 10^6$  per foot, and the total temperature is 584.7 degrees Rankine. This case is used to; (1) test the capabilities of connecting grids together where solid surfaces meet, (2) connect half grids of symmetric bodies for finite volume interpolations, (3) connect more than two grids with a general hierarchy format (Fig. 4.1).

Half body grids are used in solving flowfields which are assumed symmetric about a particular plane. The use of half grids is a valuable tool in reducing the computational time of complex flow fields. However, these grids complicate boundary connections near symmetry planes for the finite volume interpolation method. At the symmetry plane of the composite mesh, the cell



centers of different subdomain grids do not all lie within the same physical plane. Because cell center grids are used, the outer and hole boundary points near the symmetry plane need to be connected. For finite difference grids, these boundary points at the symmetry plane are taken care of by the symmetry boundary conditions. The general zeroth-order interpolation method is used to connect the cell center points at the symmetry plane, which lie outside of a connected grid. These cells reduce local order of accuracy at the symmetry plane.

The three types of half grids generated to cover the entire flow domain are: (i) Cartesian farfield grid; (ii) H-O grid around the store; (iii) O-H grid around the sting. An alternative method is to use a single grid which covers the entire flow domain. The three half grids are created separately, each using a different grid generation technique.

The H-O grid is generated around the store grid using a three step process. The physical dimensions and shape of the store cylinder are shown in Fig. 5.34. A two dimensional body fitted H-grid is generated around the store half body using the program GRAPE. Clustering is done along the body at the ogive nose cylinder and near the base of the store where the sting is connected. Next, enhanced clustering is done in the viscous region near the body using parametric curve fitting procedure as previously described. The last step is to rotate the H-grid 180 degrees, thus creating a half body three-dimensional store grid. A variable degree of rotation is used, such that in the region of the sting the  $\eta$ -lines are clustered. This allows higher probability of connecting the store and the sting grids. The store grid is shown in Fig. 5.35.

The L-shaped sting grid is generated with two constraints. These constraints are: (1) the base of the grid needs to lie completely on the surface of the store for proper grid connections and boundary conditions (see Fig.

5.36); (2) the grid needs to extend out of the Cartesian farfield because the top boundary of the sting is not defined. The physical dimensions and shape of the sting is shown in Fig. 5.34. The O-H grid is generated in several steps using simple grid techniques. A cross sectional two-dimensional O-grid is generated using the TBGG program (Fig. 5.37). The third dimension of the O-H grid is created by stacking the O-grid in the  $\zeta$ -direction. The curvature of the  $\zeta$ -planes are dependent upon the height of the sting and the radius of the store. At the sting base, the curvature of the  $\zeta$ -plane is determined by the radius of the store. Hence, the sting matches the store surface for proper grid connections. The length and width of the outer boundary of the O-grid increase linearly with the height of the sting. This is done to create a larger physical space for the overlapped region (Fig. 5.38).

The Cartesian farfield mesh is generated using simple algebraic techniques. The dimensions of each subdomain grid are given in Table 5.1.

The composite mesh is generated using the program MaGGiE (Fig. 5.39). A general composite grid hierarchy is used in this case. The three grids are connected to each other. The Cartesian farfield grid is the global grid, denoted as  $G_{1,1}$ . The store grid and the sting grid are on the same composite grid level denoted by  $G_{2,1}$  and  $G_{2,2}$ , respectively. The grids  $G_{2,1}$  and  $G_{2,2}$  are connected to each other and to grid  $G_{1,1}$ . The store grid creates a hole in the Cartesian grid. The sting grid creates a hole in the store and Cartesian grids. When the sting creates the hole in the store grid, a hole is generated on the surface of the store body since they are connected. Hence, the boundary conditions at the connected surface can be specified completely within the sting grid for the flow solver. The hole boundary cells surrounding the store are shown in Fig. 5.40. The overlapped region between the store and Cartesian grids is eight cells (Fig. 5.41). The overlapped region between the sting and store grids is also

eight cells (Fig. 5.42). The overlapped region between the sting and Cartesian grid is 15 cells. The larger overlapped region is necessary because the Cartesian grid is not clustered in the region of the sting. The Cartesian cell sizes are larger than the cell sizes of the sting grid. From the viewpoint of the accuracy of interpolation, this is not desirable, because the interpolation is dependent on distance between cells and not the weighted cell volumes. The Cartesian grid clustering is generated for the purpose of connecting to a rectangular cavity grid for future work. The number of hole cells and boundary cells for each mesh is given in Table 5.1.

A total of 220 cells, out of 11,029 hole and outer boundary cells, use zeroth-order interpolation method, instead of the trilinear interpolation method for connections. This is less than two percent of the boundary cells. These cells are located at the symmetry plane. This is expected, because the cell center grids do not align themselves with each other at the symmetry plane of the composite mesh.

The flow solutions of the composite half body grids are presented in Figs. 5.43-45. The subdomain grids are initialized by the freestream conditions. Initially, the subdomain grids of the store and sting are run separately, but instabilities in the solution occur, because of the small physical domain sizes of the grids. Hence, their solutions are not used to initialize the composite mesh. Finest level calculation, without mesh sequencing and multigriding, is used to develop the composite flow field. Shown in Fig. 5.43 are the longitudinal density contours at the symmetry plane of the composite mesh. The contour lines pass smoothly across the interfaces. There are no artificial shocks created at the boundaries. At the base of the store the expansion and compression waves pass also smoothly across the interface. A complicated flow occurs behind the sting, where the base flow of the sting interferes with the

base flow of the store. There are interactions of compression and expansion waves. A top view of the longitudinal density contours around the sting is shown in Fig. 5.44. At the interface between the sting and Cartesian grids, the compression and expansion waves cross smoothly without distortions. A three dimensional composite view of the density contours is shown in Fig. 5.45. This figure represents the type of complex flowfields that can be accommodated by the overlapping/embedding method.

#### 5.6 Case 6: Store with Fins and a Curved Sting Near a Cavity

A composite mesh is generated for a configuration of a store with fins and a curved sting positioned above a rectangular cavity. There are five subdomain half body grids within the composite mesh. The five half grids are: (1) a farfield Cartesian grid above a cavity; (2-4) three H-O zonal grids around a store with fins and a sting; (5) a Cartesian zonal cavity grid. This case is used to demonstrate the overlapping/embedding capabilities of MaGGiE for a composite grid made up of more than three grids using zonal and overlapped half body grids.

Zonal grids are grids that are patched together along a constant surface. Across the zonal interface the grid lines can be discontinuous or continuous depending on the conservative treatment at the boundary. The zonal grids used in this example are one to one cell matching, hence the grid lines are continuous across the interface. One to one cell matching is the best method for conserving fluxes across boundaries.

The two Cartesian grids are generated using simple algebraic techniques with parametric and exponential clustering. The cavity grid is connected to the farfield grid at the first  $\zeta$ -constant plane. There is one to one matching of cell centers at the zonal plane (Fig. 5.46). An alternative method of connecting the two grids is to overlap them. However, one to one matching is done at the

interface to conserve the fluxes exactly. This is important for cavity flow, where there is an unsteady shear layer propagating across the top of the cavity and mass being pumped into and out of the cavity. Hence, although the cavity is included within the composite mesh, the grid is not connected to any other grid by the overlapping procedure.

Each of the three zonal H-O store grids with fins and extended sting is generated using the three dimensional surface grid generation code EAGLE [49,50]. The combination of the three grids create the half store body (Fig. 5.47). The three zones are shown in a cross sectional view of the store in Fig. 5.48. The zonal  $\eta$ -planes define the fin surfaces. There is a one to one matching of cell centers across the zonal surfaces above the fins.

The composite mesh is generated using the MaGGiE code. A simple composite overlapped hierarchy is used in this case. The three store grids are connected to the Cartesian farfield grid only. The Cartesian farfield grid is the global grid, denoted as  $G_{1,1}$ . The three store grids are on the same composite grid level denoted by  $G_{2,1}$ ,  $G_{2,2}$ , and  $G_{2,3}$ , respectively. The cavity grid is considered to be on another composite grid level,  $G_{3,1}$ . However, this grid is not being used in the grid connection scheme. The three store grids create a hole in the Cartesian farfield grid away from the fins (Fig. 5.49). To create such a hole using the zonal grids, a simple modification is done. The three grids are combined into one grid to create the hole. Next, they are separated to obtain proper grid connections to the hole and outer boundary cells. This modification is needed because the specified hole boundaries around the zonal grids cannot be defined properly to be used in the hole cells search method described in Section 4.2.1. Shown in Fig. 5.50 is the hole boundary cells of the Cartesian mesh that are connected to the store grids. There is a 15-cell overlapped region between the store grids and the Cartesian mesh. The 15 cell

overlap is needed because the Cartesian cell sizes are much larger than the store cell sizes in this region. Figure 5.51 shows the overlapped region in the longitudinal plane of the composite mesh. The number of hole cells and boundary cells for each subdomain mesh is given in Table 5.1.

A total of 457 cells, out of 10,805 hole and outer boundary cells, use the zeroth-order interpolation method for connections. These cells are located near or at the symmetry plane of the composite mesh. This is expected because the cell center grids' symmetry planes do not lie in the same physical plane. This phenomena is also noted in Case 5, where half body grids are also used.

### 5.7 Comparisons and Comments

The first three cases, used in validating the overlapping/embedding procedure for finite volume multigrid levels, provide valuable insight into understanding the problems of conserving fluxes across interfaces. The results of these cases compare well with the solution obtained on the single-domain BNC grid. The jumps in contour lines across the interfaces is considered mostly due to inadequacies of the plotting procedure and slightly due to a loss of conservation.

The last three cases provide a variety of different problems that can occur when creating a composite mesh of cell centers for a real configuration. The sting that is associated with each store is included as part of the configuration to represent actual experimental tests. The stings are used to support the stores within the wind tunnels. The problems that are dealt with are: (1) bodies in close proximity of each other; (2) half body connections at symmetry plane; (3) zonal grids with overlapped grids; (4) general overlapped and simple overlapped hierarchial connections for a composite mesh.

Iterations of the overlapping/embedding method is needed for each case. The iterations are used to define hole and outer boundaries of each subdomain

for proper grid connections. However, these iterations can be eliminated, if careful consideration of grid sizes and clustering within the overlapped region are done. Overall, creating the composite meshes is straightforward. The composite mesh connections are dependent upon the generation of the subdomain grids to a certain degree. However, the subdomain grids are generally of simple topology.

## Chapter 6

### CONCLUSIONS

A domain decomposition method, called grid overlapping/embedding, is used in simplifying computations on three-dimensional complex configurations. The overlapping/embedding method divides the flowfields into simpler subdomains. These domains are either completely embedded within each other, or simply overlapped. The overlapping procedure is developed to create composite meshes using cell center grids for finite volume solution algorithms. A procedure is developed for overlapping coarser level grids in a composite mesh for multigrid solution algorithms. The product of this investigation is a computer code, given the name MaGGiE, (Multi-Geometry Grid Embedder). MaGGiE is developed to take independently generated component grids and their overlapping structure as input, and it creates a composite mesh and interpolation data to be used by a finite volume solution algorithm with or without multigriding.

The overlapping method is applied successively to six composite grid cases.

The conclusions are listed, which are drawn from the experiences with these cases, may be outlined as follows:

- (1) The subdomain grids of the composite meshes are easily generated using simple grid techniques.
- (2) Finite volume grid connections are made for all six cases. The solutions obtained on the composite meshes compare well with the



experimental, analytical, and single-domain calculations, where applicable. The longitudinal pressure contours and cross flow density contours for the Cases 1-3 compare well with the solution on the single-domain grid. A slight loss of conservation is noticeable near the hole boundary in the crossflow density contours for the simply overlapped Case 3. Although there are no experimental data for the composite grid solutions of the ONC case or for the store/L-shaped sting case, their solutions are considered reasonable for such complex flows.

- (3) The nonconservative trilinear interpolation method, which is used with the Roe flux-splitting scheme, transfers the time fluxes across the mesh boundaries with little loss of conservation. No artificial shocks are created at the boundaries. Compression and expansion waves pass across the interfaces with little dispersion.
- (4) The multigrid connections are implemented for the composite grid case of the BNC embedded within a Cartesian farfield. The coarser level grids are easily connected. The convergence rate of two-level multigrid computations is about 0.98 as opposed to 0.99, which is obtained without multigridding..
- (5) An overlapped region of 5 to 15 cells is found to be adequate for proper grid connections without redundant information being passed between subdomains. However, a parametric study of the optimum width of the overlapped region has not been done.
- (6) The grid connections on the coarser grid levels, excluding the illegal zones, are independent of the overlapped regions of the finest level meshes.

- (7) Zonal grids are incorporated within the composite mesh of Case 6 with little modification to MaGGiE.
- (8) Introducing the zeroth-order interpolation procedure increases the robustness of the grid overlapping/embedding method for cell center grids, specifically, at outer boundaries, such as, symmetry planes and surface contacts.

The recommendations for future work are listed below.

- (1) The dependence of the rate of convergence of the solution on the width of the overlapped regions, should be investigated.
- (2) The conservation across the interfaces using a wider interpolation stencil, i.e. increasing the set of hole and boundary cells to two, should be investigated.
- (3) The MaGGiE code should be optimized in order to integrate the code within a solution algorithm for dynamic grids.
- (4) An accurate method of plotting composite grid solutions should be developed, for better determination of interface interactions at nonmatching grid planes.

## REFERENCES

1. Ghia, K.N., and Ghia, U., Advances in Grid Generation, Applied Mechanics, Bioengineering and Fluids Engineering Conference, ASME FED-Vol. 5, June 1983, pp. iii-v
2. Nakahashi, K., and Obayashi, S., "FDM-FEM Zonal Approach for Viscous Flow Computations Over Multiple-Bodies," AIAA 87-0604, January 1987.
3. Hennesius, K. A., and Pulliam, T. H., "A Zonal Approach to Solution of the Euler Equations," AIAA 82-0969, June 1982.
4. Rai, M.M., "A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations," AIAA 84-0164, January 1984.
5. Hennesius, K.A., and Rai, M.M., "Three-Dimensional, Conservative, Euler Computations Using Patched Grid Systems and Explicit Methods," AIAA 86-1081, May 1986.
6. Kathong, M., Application of Advanced Grid Generation Technique For Flow Fields About Complex Configuration, Ph.D. Dissertation, Mechanical Engineering and Mechanics Department, Old Dominion University, August 1988.
7. Ramshaw, J.D., "Conservative Rezoning Algorithm for Generalized Two-Dimensional Meshes," Journal of Computational Physics, vol. 59, June 1985, pp. 193-199
8. Thomas, J.L., Walters, R. W., Reu, T., and Ghaffari, F., Weston, R.P., Luckring, J.M., "A Patched-Grid Algorithm for Complex Configurations Directed Towards the F/A-18 Aircraft," AIAA 89-0121, January 1989.
9. Nakahashi, K., and Obayashi, S., "Viscous Flow Computations Using a Composite Grid," AIAA 87-1128, July 1987.

10. Atta, E., "Component-Adaptive Grid Interfacing," AIAA 81-0382, January 1981.
11. Atta, E. H., and Vadyak, J., "A Grid Interfacing Zonal Algorithm for Three Dimensional Transonic Flows About Aircraft Configurations," AIAA 82-1017, June 1982.
12. Steger, J.L., Dougherty, F.C., and Benek, J.A., "A Chimera Grid Scheme," Advances in Grid Generation, FED-Vol. 5, 1983, pp. 59-69.
13. Benek, J.A., and Steger, J.L., and Dougherty, F.C., "A Flexible Grid Embedding Technique with Application," AIAA 83-1944, July 1988.
14. Benek, J.A., and Buning, P.G., and Steger, J.L., "A 3-D Chimera Grid Embedding Technique," AIAA 85-1523, July 1985.
15. Benek, J.A., and Donegan, T.L., and Suhs, N.E., "Extended Chimera Grid Embedding Scheme with Application to Viscous Flows," AIAA 87-1126, 1987.
16. Suhs, N.E., "Computations of Three-Dimensional Cavity Flow at Subsonic and Supersonic Mach Numbers," AIAA 87-1208, June 1987.
17. Dougherty, F.C., and Kuan, J-H., "Transonic Store Separation Using a Three-Dimensional Chimera Grid Scheme," AIAA 89-0637, January 1989.
18. Dougherty, F.C., Benek, J.A., and Steger, J.L., "On Applications of Chimera Grid Schemes To Store Separation," NASA Technical Memorandum 88 1936, October 1985.
19. Chessire, G., and Henshaw, W.D., "Composite Overlapping Meshes for the Solution of Partial Differential Equations," IBM Research Report, RC 14355 (#64300), Mathematics Research Report, January 1989.
20. Chessire, G., and Henshaw, W.D., "Multigrid on Composite Meshes, " SIAM Journal Scientific Statistical Computations, Vol. 8, No. 6, November 1987. pp. 914-923.
21. Baysal, O., Fouladi, K., and Miller, D.J., "Computations of Supersonic Flows over a Body at High Angles of Attack," AIAA Journal, Vol. 27, No. 4, April 1989, pp. 427-437.

22. Baldwin, B.S., Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA 78-257, January 1978.
23. Anderson, W.K., Thomas, J.L., and Whitfield, D.L., "Three-Dimensional Multigrid Algorithms for the Flux-Split Euler Equations," NASA Technical Paper 2829, November 1988.
24. Mulder, W.A., and van Leer, B., "Implicit Upwind Methods For the Euler Equations," AIAA 83-1930, January 1983.
25. Hanel, D, Schwane, R., and Seider, G., "On the Accuracy of Upwind Schemes for the Solution of the Navier-Stokes Equations," AIAA 87-1105, January 1987.
26. Anderson, D.A., Tannehill, J.C., and Pletcher, R.H., Computational Fluid Mechanics and Heat Transfer, Hemisphere Publishing Corp., New York, 1984., pp.260-261.
27. Roe, P.L., "Characteristic-Based Schemes for Euler Equations," Annual Review of Fluid Mechanics, Vol. 18, 1986, pp 337-365
28. Pulliam, T.H., and Chaussee, D.S., "A Diagonal Form of an Implicit Approximate Factorization Algorithm," Journal of Computational Physics, Vol. 39, 1981, pp.347-363.
29. Hackbusch, W., Multi-Grid Methods and Applications, Springer-Verlag, New York, 1985
30. Brandt, A., "Guide to Multigrid Development, Multigrid Methods," Lecture Notes in Mathematics, Vol. 960, Springer-Verlag, New York, 1982, pp. 220-312.
31. Brandt, A., "Multi-Level Adaptive Solutions to Boundary-Value Problems," Mathematical Computations, Vol. 31, No. 138, April 1977, pp.333-390.
32. Anderson, W.K., and Thomas, J.L., "Multigrid Acceleration of the Flux Split Euler Equations," AIAA 86-0274, January 1986.

33. Baysal, O., "Supercomputing Supersonic Flows Using Upwind Relaxation and MacCormack Schemes," ASME Journal of Fluids Engineering, Vol. 110, No. 1, March 1988, pp.62-68.
34. Smith, R.E., and Weigel, B.L., "Analytic and Approximate Boundary Fitted Coordinate System for Fluid Flow Simulation," AIAA 80-0192, 1980.
35. Mastin, C.W., and McConnaughey, H.V., "Composite Problems on Composite Grids," AIAA 84-1611, June 1984.
36. Smith, R.E., and Wiess, M.R., "An Interactive Algebraic Grid Generation Technique," NASA TP-2533, March 1986
37. Steger, J.L., and Sorenson, R.L., "Automatic Mesh-Point Clustering Near a Boundary in Grid Generation with Elliptic Partial Differential Equations," Journal of Computational Physics, Vol. 33, No. 3, December 1979, pp. 405-410.
38. Sorenson, R.L., "A Computer Program to Generate 2-D Grids About Airfoils and Other Shapes by the Use of Poisson's Equations," NASA Technical Memorandum 81198, NASA Ames Research Center, 1988.
39. Chung, T.J., Finite Element Analysis in Fluid Dynamics, McGraw-Hill, 1978
40. Heubner, K.H., and Thornton, The Finite Element Method for Engineers, 2nd Edition, A Wiley-Interscience Publication, New York, 1982.
41. Ergatoudis, J.G., Irons, B.M., and Zienkiewics, O.C., "Curved Isoparametric Quadrilateral Elements for Finite Element Analysis," International Journal of Solids Structure, Vol. 4, 1968, pp. 31-42.
42. Berger, M.J., "On Conservation at Grid Interfaces," SIAM Journal of Numerical Analysis, Vol. 24, No. 5, October 1987, pp. 968-983.
43. Walters, R.W., Reu, T., McGrory, W.D., and Thomas, J.L., Richardson, P.F., "A Longitudinally-Patched Grid Approach with Applications to High Speed Flows," AIAA 88-0715, January 1988.
44. Baysal, O., Fouladi, K., and Lessard, V.R., "A Multigrid and Upwind Viscous Flow Solver on 3-D Embedded and Overlapped Grids," AIAA 89-0464, January 1989.

45. Fouladi, K, Navier-Stokes Solutions About Complex Configurations Using Domain Decomposition Techniques, Ph. D. Dissertation (in preparation), Mechanical Engineering and Mechanics Department, Old Dominion University, 1990.
46. Landrum, E.J., and Babb, C.D., "Wind-Tunnel Force, Pressure and Flow Visualization Data at Mach Numbers from 1.6 to 4.63 for a Series of Bodies of Revolution at Angles of Attack from -4 to 60 degrees," NASA TM X-3558, October 1977 and NASA TM-78813, March 1979
47. Belk, D.M., and Whitfield, D.L., "Time-Accurate Euler Equations Solution on Dynamic Blocked Grids", AIAA 87-1127, June 1987.
48. Chow, R., and Marconi, F., "A Navier-Stokes Solution to Hypersonic Store Separation Flow Fields", AIAA 89-0031, January 1989.
49. Thompson, J.F., Lijewski, L.E., and Gatlin, B., "Efficient Application Techniques of the EAGLE Grid Code to Complex Missile Configurations", AIAA 89-0361, January 1989.
50. Leung, R.W., and Sheftic, J.S., Private Communications and Cooperations., Lockheed Missles and Space Co., Inc., Sunnyvale, CA 1989.

	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	CASE 6
Grid Hierarchy No. Grid Topology Grid Level Grid Dimensions (i,j,k) Physical Domain (i,j,k),D No. of Hole Cells No. of Boundary Cells Overlapped Cells	1 Cartesian (H-H) 81, 73, 73 53, 40, 40 27906 7296 10	1 Cartesian (H-H) 2 81, 65, 25 53, 40, 40 652 4234 N/A	1 C-O 81, 65, 25 37, 20, 20 5264 5266 8	1 Cartesian (H-H) 1 97, 57, 73 53, 36, 36 1920 1210 7	1 Cartesian (H-H) 1 99, 28, 57 34, 4, 9 9540 2684 8	1 Cartesian (H-H) 1 108, 21, 50 12, 2, 2, 5 7580 2594 15
Grid Hierarchy No. Grid Topology Grid Level Grid Dimensions (i,j,k) Physical Domain (i,j,k),D No. of Hole Cells No. of Boundary Cells Overlapped Cells	2 C-O 81, 65, 45 37, 4, 4 0 5265 10	2 C-O 2 41, 33, 23 37, 4, 4 0 357 N/A	2 C-O 1 81, 65, 41 37, 3, 5, 3, 5 0 5265 8	2 C-O 1 73, 65, 17 37, 6, 6 0 4680 7	2 H-O 1 121, 29, 33 24, 3, 3 10440 5211 8	2, 4 H-O 1 121, 13, 41 75, 1, 1 0 2093 15
Grid Hierarchy No. Grid Topology Grid Level Grid Dimensions (i,j,k) Physical Domain (i,j,k),D No. of Hole Cells No. of Boundary Cells Overlapped Cells	N/A	N/A	N/A	N/A	3 O-H 1 21, 25, 57 2, 5, 3, 4 0 1134 8	3 H-O 1 121, 25, 41 75, 1, 1 0 4025 15
Grid Hierarchy No. Grid Topology Grid Level Grid Dimensions (i,j,k) Physical Domain (i,j,k),D No. of Hole Cells No. of Boundary Cells Overlapped Cells	N/A	N/A	N/A	N/A	N/A	5 Cartesian (H-H) 1 91, 16, 30 1, 0, 1, 1 0 0 0

No. of Composite Cells      668574      87248      347490      484282      303726      410091

Table 5.1 Summary of composite mesh cases



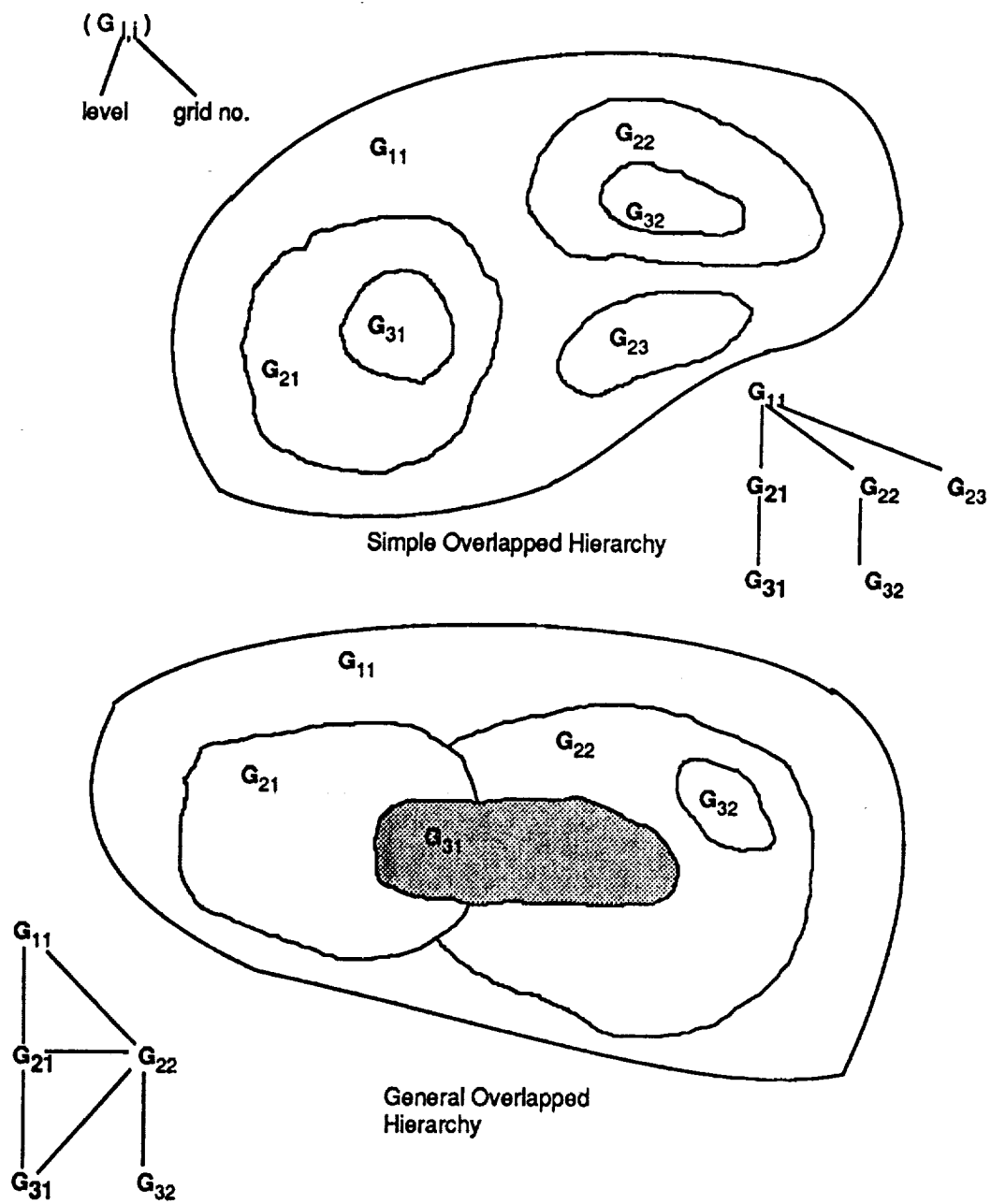


Figure 4.1 Hierarchy ordering for intergrid communications

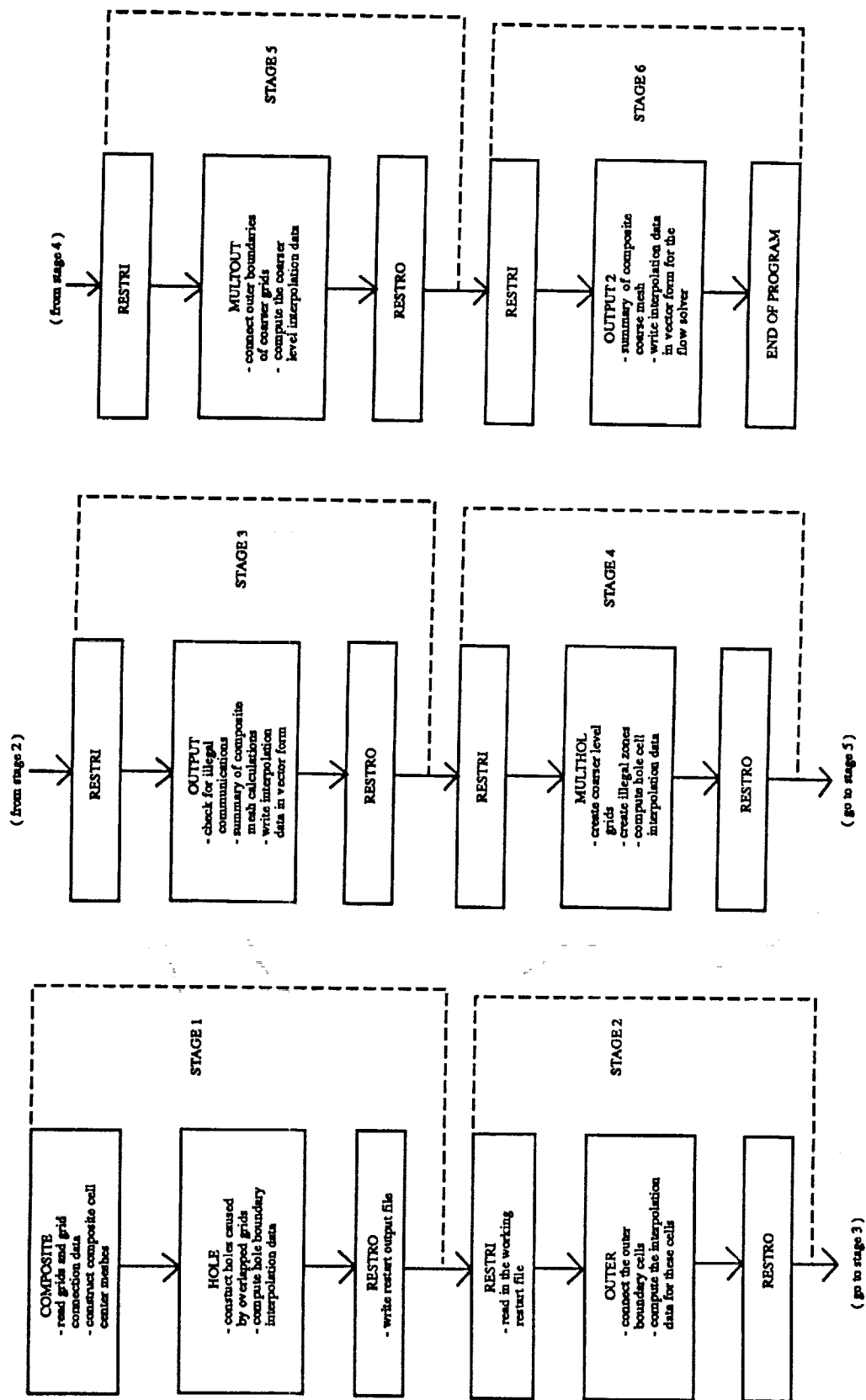


Figure 4.2 Overview flow chart of the MaGGiE program

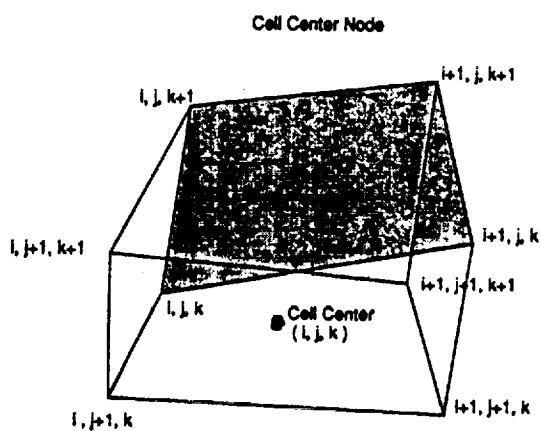


Figure 4.3 Sketch of a cell center node

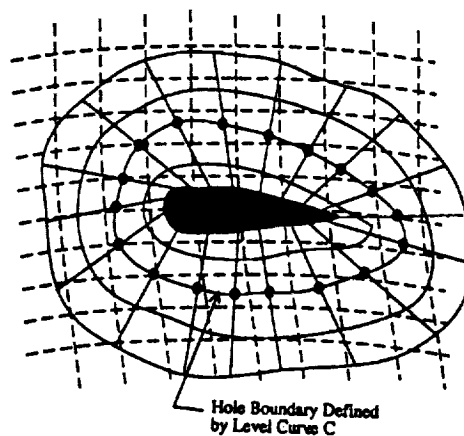


Figure 4.4 Sketch of an initial composite mesh hole boundary surface

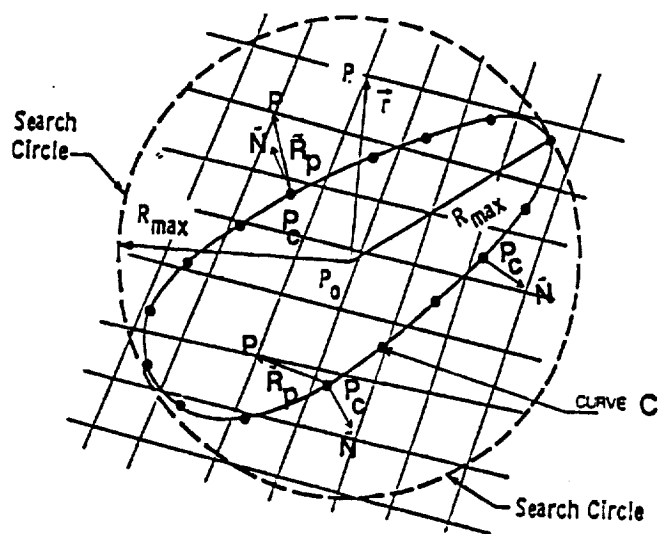


Figure 4.5 Sketch of hole search method

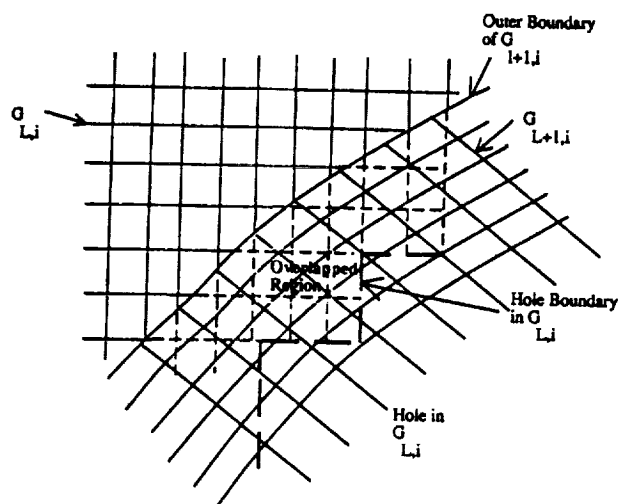


Figure 4.6 Sketch of the hole and outer boundary of a composite mesh

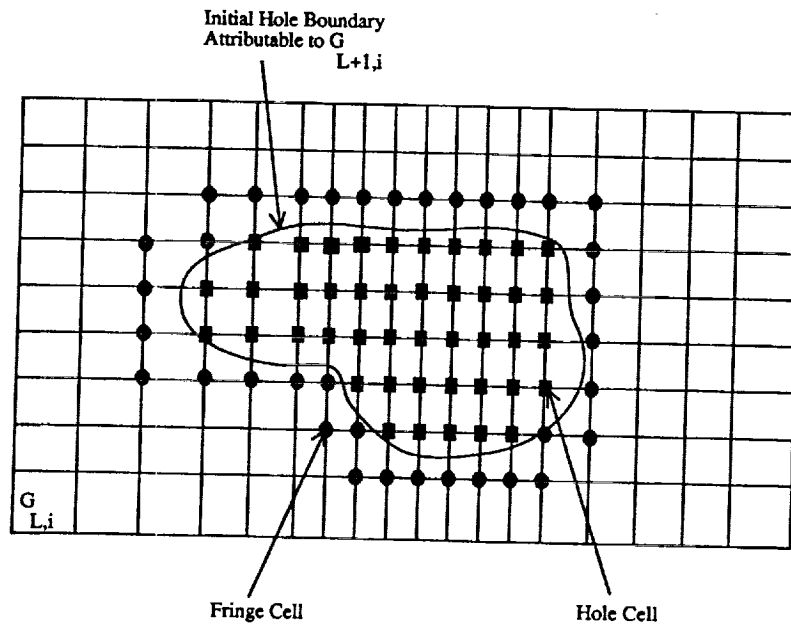


Figure 4.7 Sketch of fringe and hole cells in a composite mesh

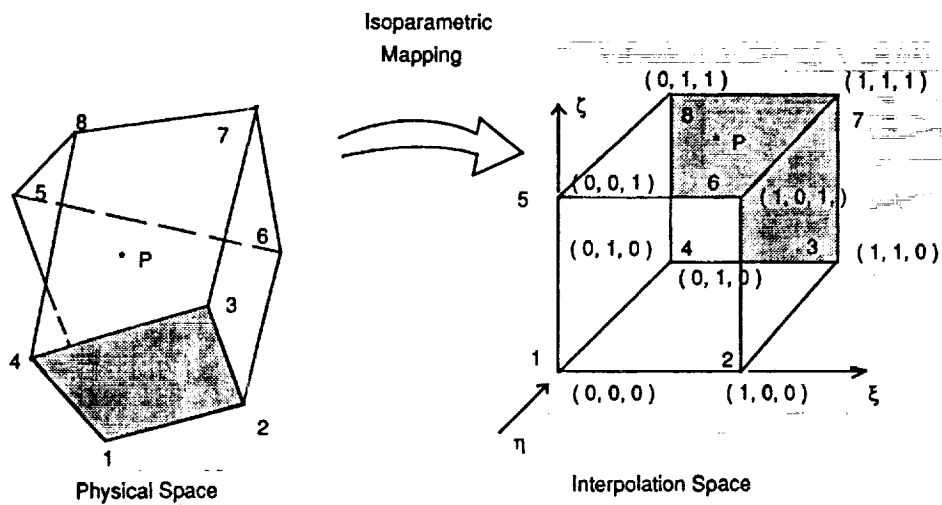
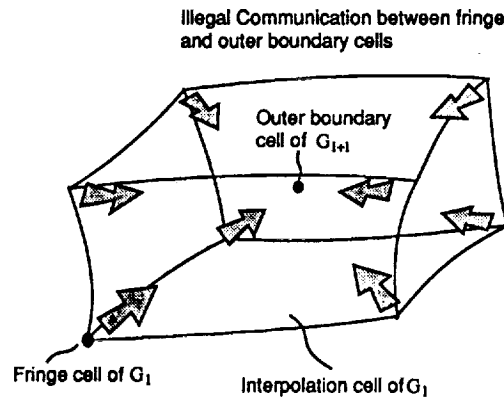


Figure 4.8 Sketch of isoparametric mapping



ORIGINAL PAGE IS  
OF POOR QUALITY.

Figure 4.9 Sketch of illegal communication between subdomains

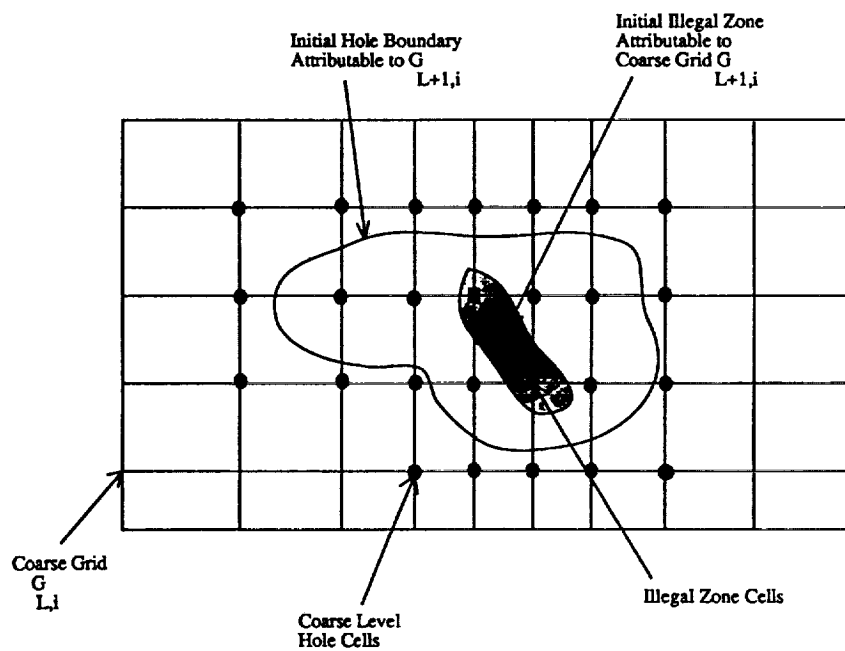


Figure 4.10 Sketch of an illegal zone on a composite coarse level mesh

$a_{11}$	$a_{12}$	0	0	0	0	0	0	0	$\Delta Q$	=	$R$
$a_{21}$	$a_{22}$	$a_{23}$	0	0	0	0	0	0	$\Delta Q$		$R$
0	$a_{32}$	$a_{33}$	$a_{34}$	0	0	0	0	0	$\Delta Q$		$R$
0	0	0	I	0	0	0	0	0	$\Delta Q$		0
0	0	0	0	I	0	0	0	0	$\Delta Q$		0
0	0	0	0	0	I	0	0	0	$\Delta Q$		0
0	0	0	0	0	0	$a_{76}$	$a_{77}$	$a_{78}$	$\Delta Q$		$R$
0	0	0	0	0	0	0	$a_{87}$	$a_{88}$	$\Delta Q$		$R$
0	0	0	0	0	0	0	0	$a_{89}$	$\Delta Q$		$R$
0	0	0	0	0	0	0	0	$a_{98}$	$\Delta Q$		$R$
								$a_{99}$			

HOLE

Figure 4.11 Solution on a grid which accommodates a hole

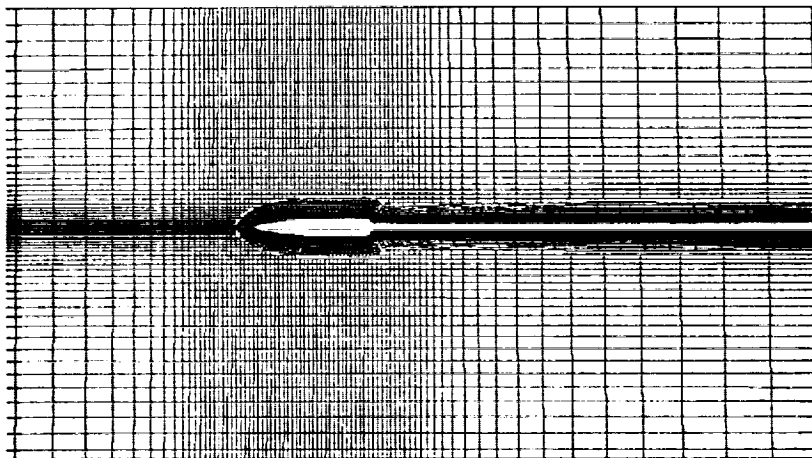


Figure 5.1 Composite grid of a blunt-nose cylinder in a Cartesian mesh

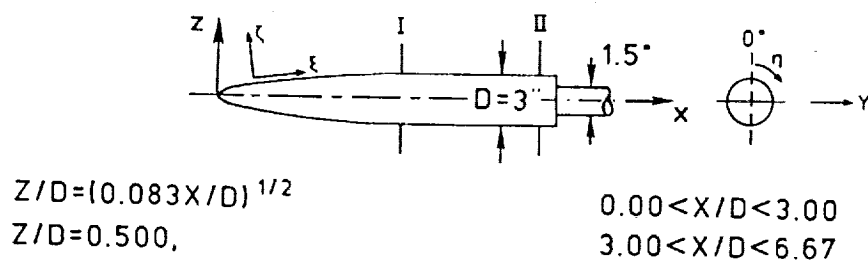


Figure 5.2 Schematic of the blunt-nose cylinder (BNC)

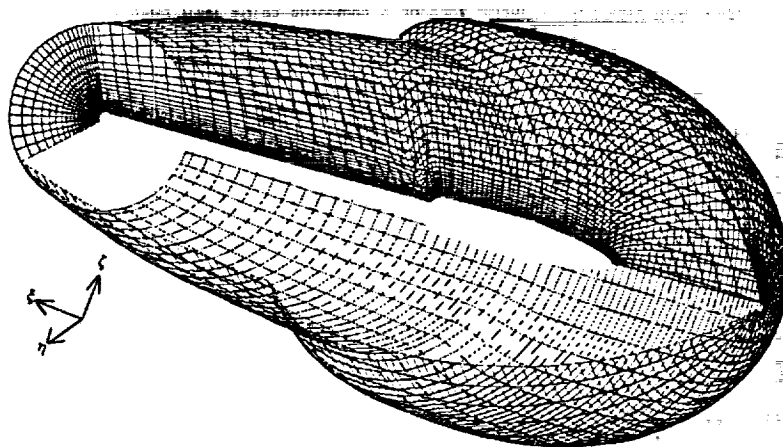


Figure 5.3 Plot of the BNC C-O grid

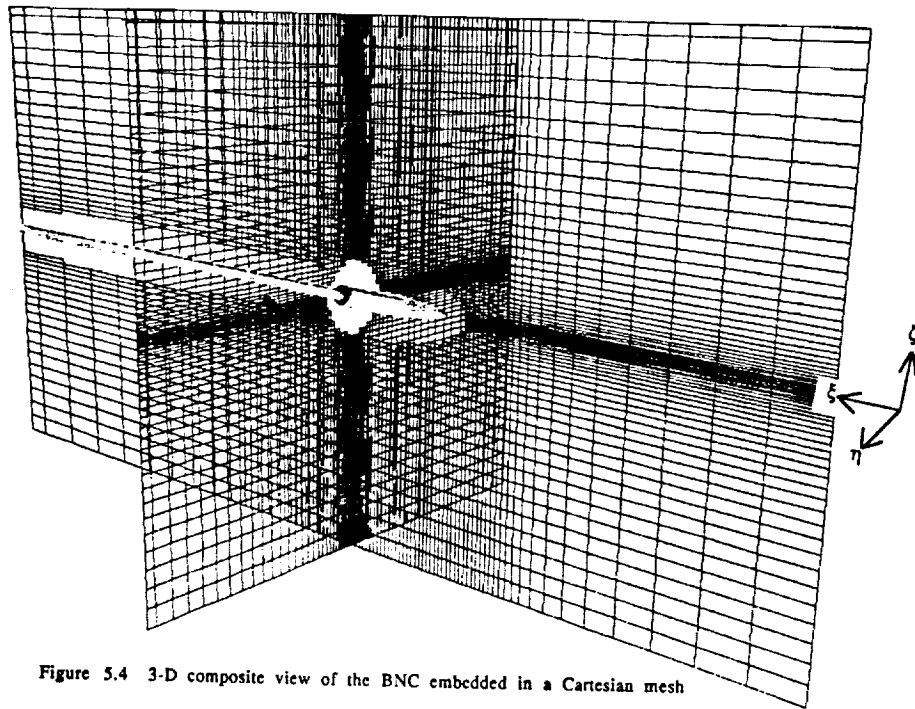


Figure 5.4 3-D composite view of the BNC embedded in a Cartesian mesh

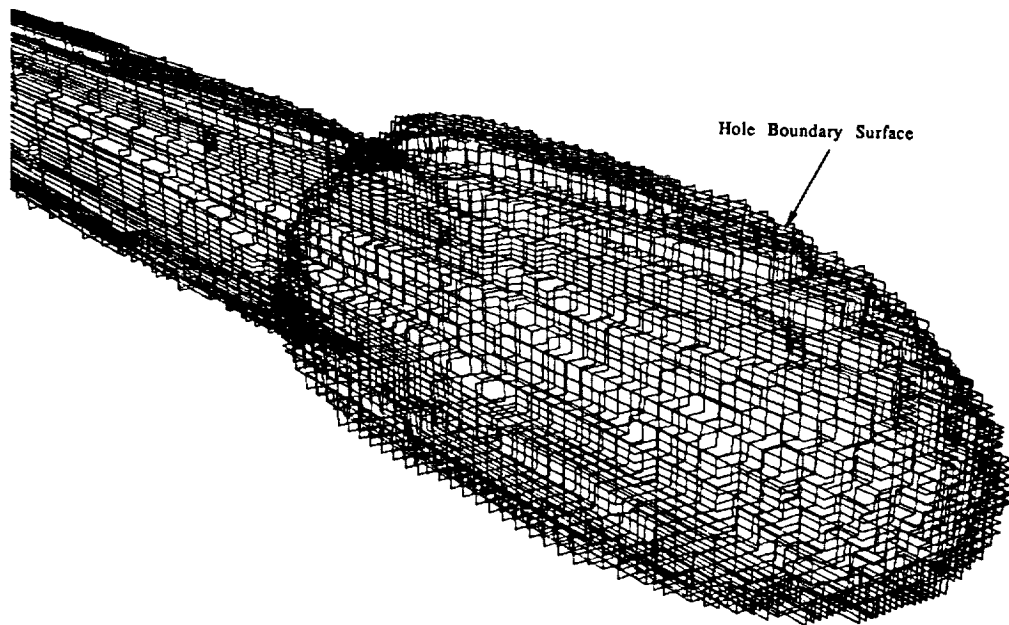


Figure 5.5 The hole boundary in the Cartesian Mesh created by the BNC

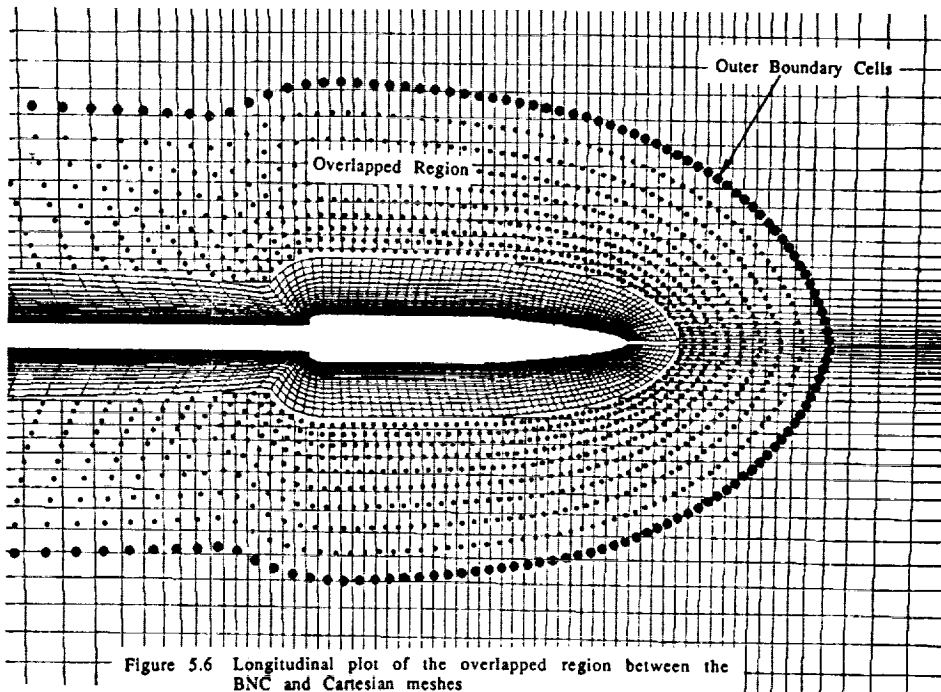


Figure 5.6 Longitudinal plot of the overlapped region between the BNC and Cartesian meshes

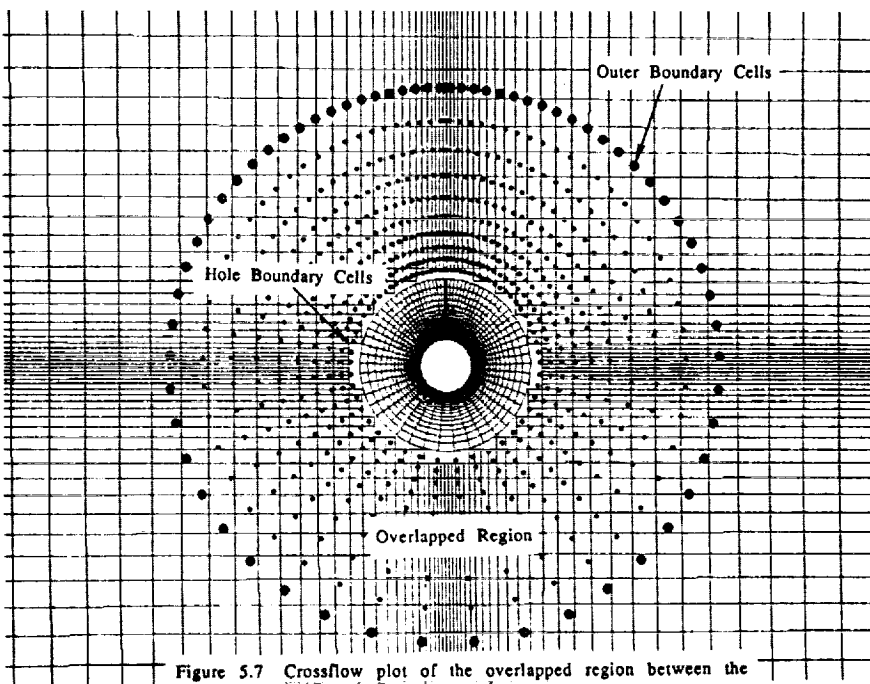


Figure 5.7 Crossflow plot of the overlapped region between the BNC and Cartesian meshes



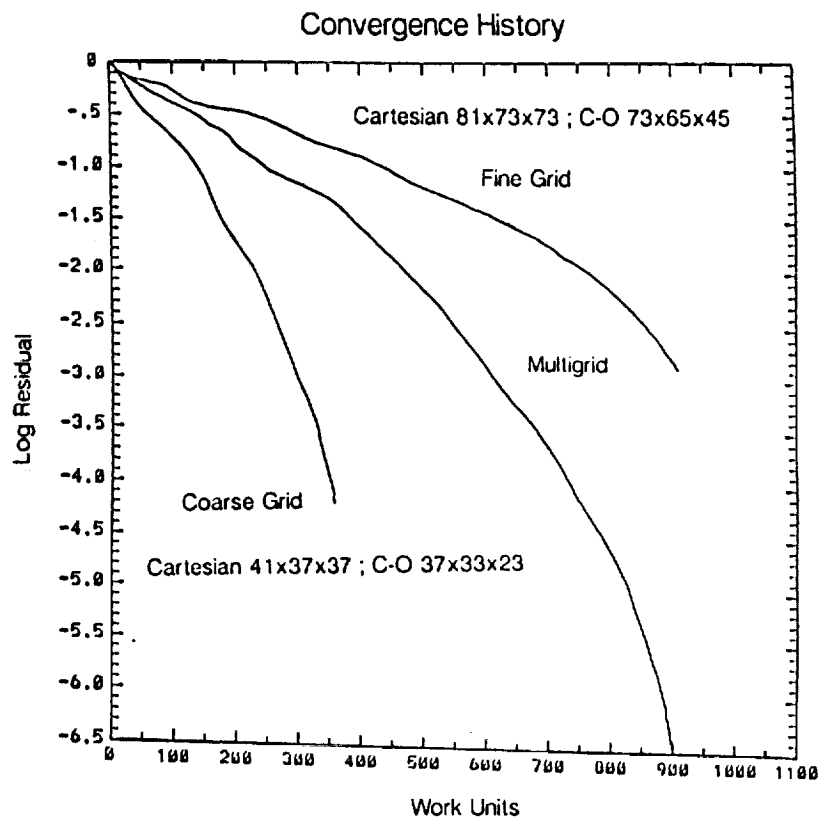


Figure 5.8 Residual history of the composite BNC and Cartesian solution

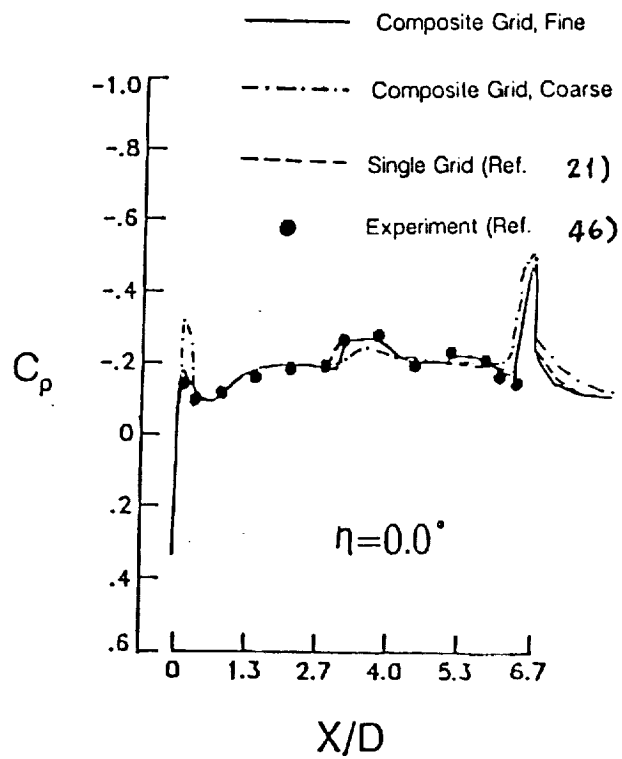


Figure 5.9 Longitudinal pressure coefficient on the leeward side of the composite BNC mesh

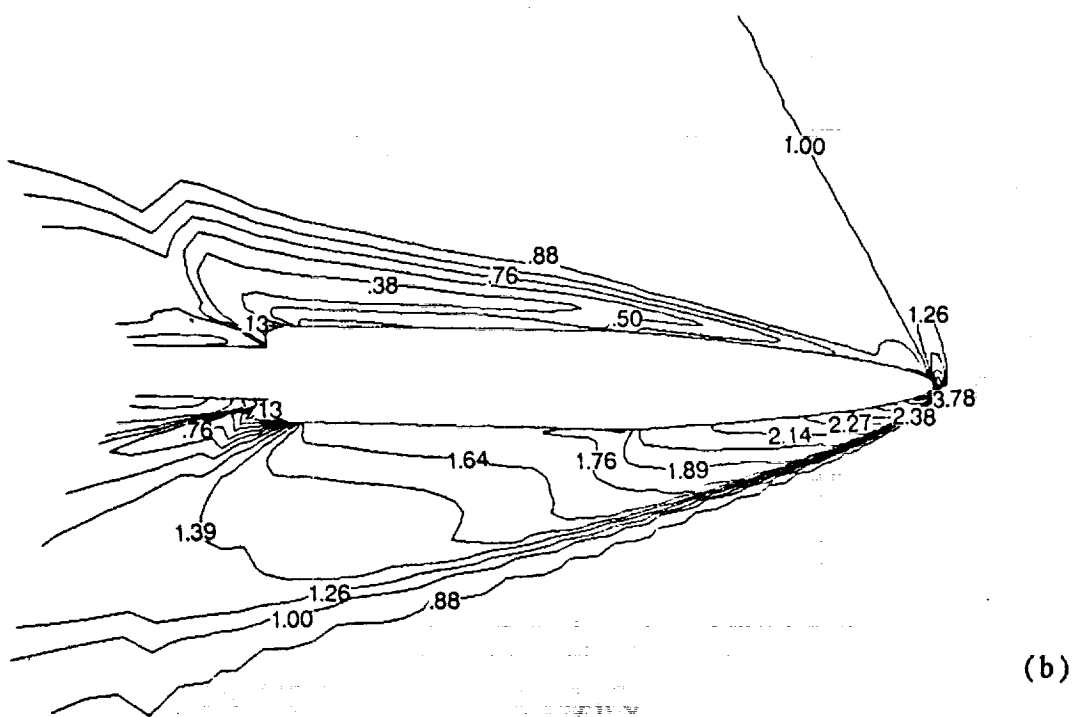
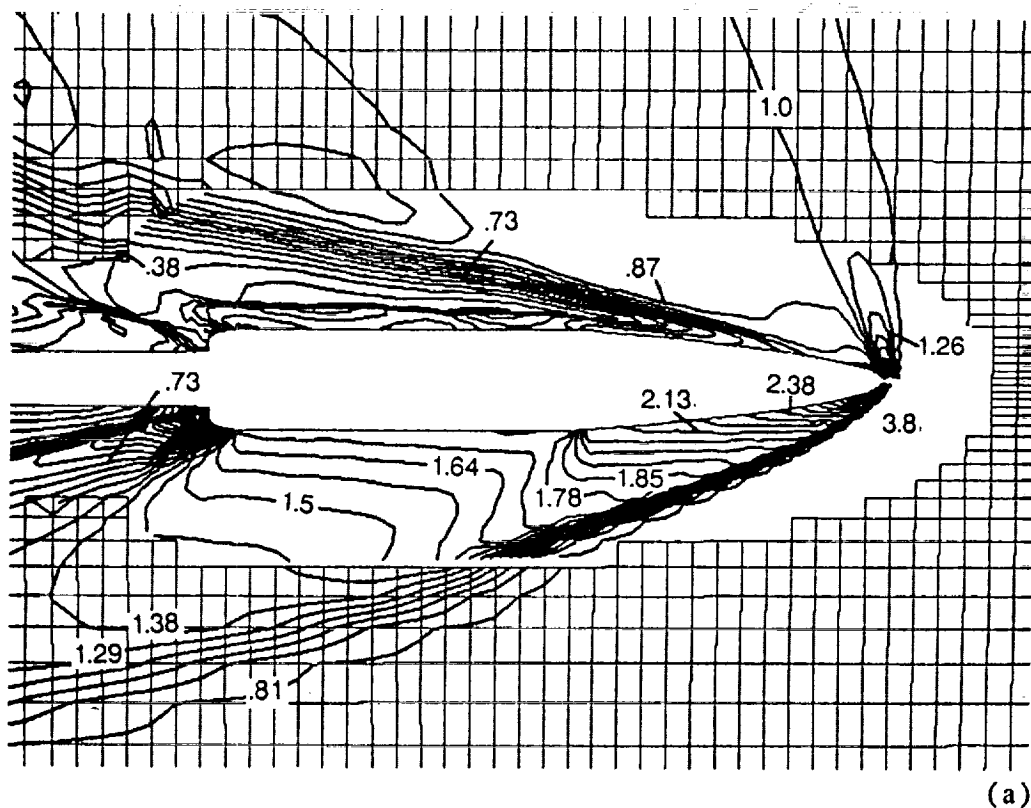


Figure 5.10 Normalized pressure contours on the symmetry plane of the BNC computed on a) the composite grid, b) the single C-O grid without embedding

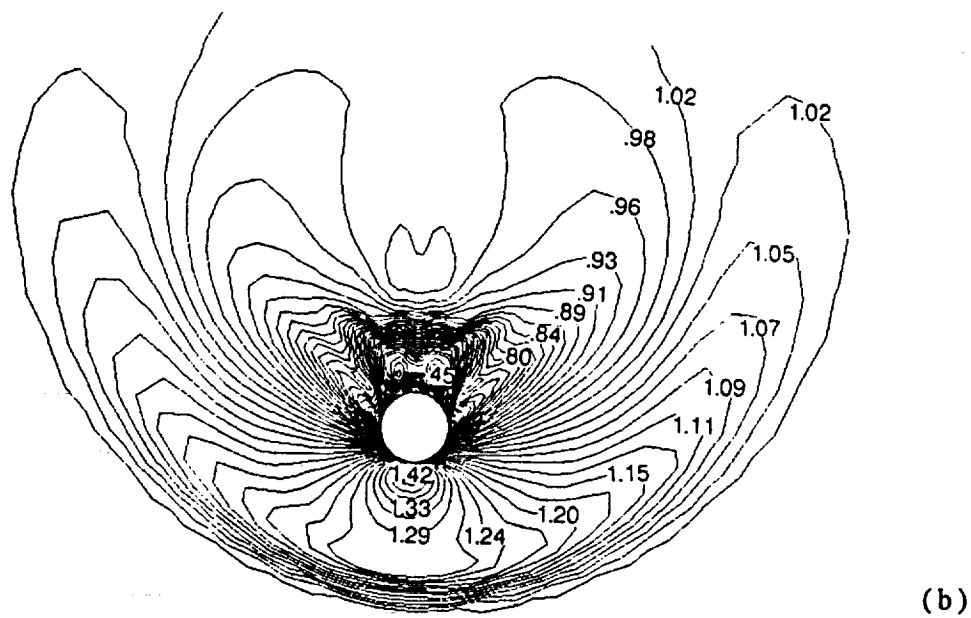
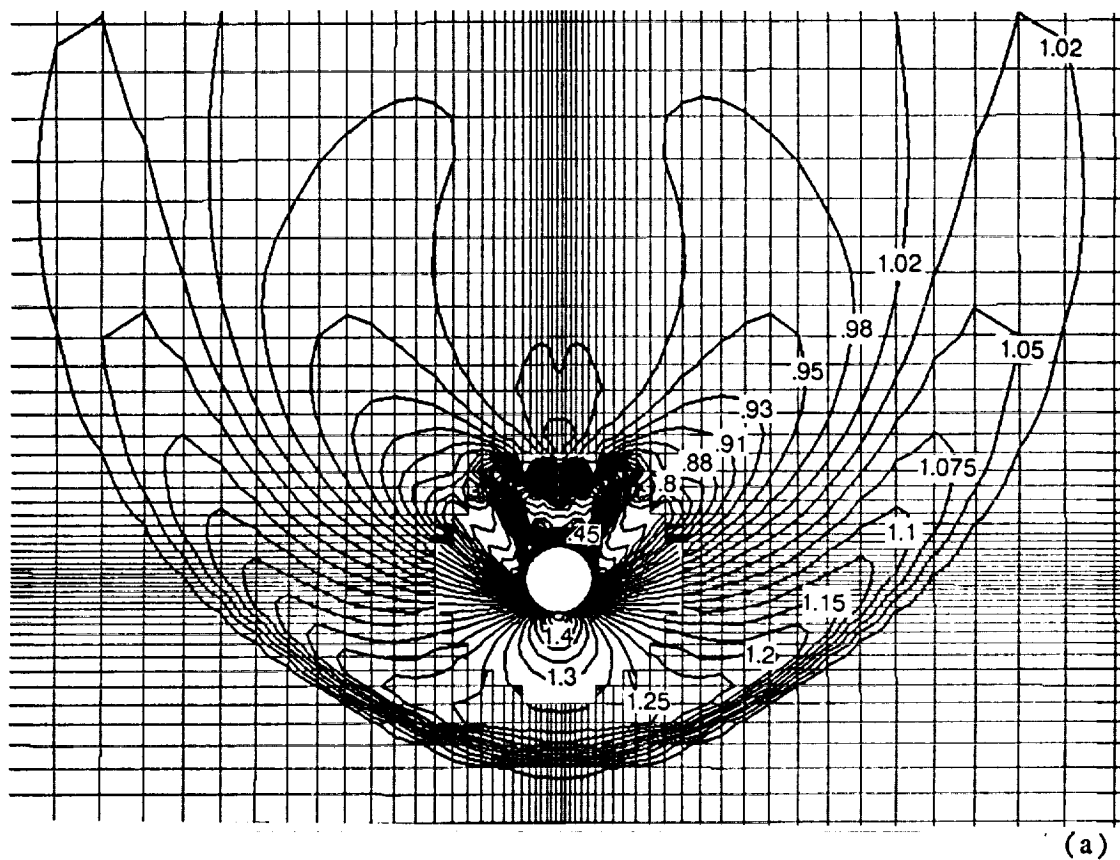


Figure 5.11 Normalized density contours on the crossflow plane of the BNC computed on a) the composite grid, b) the single C-O grid without embedding

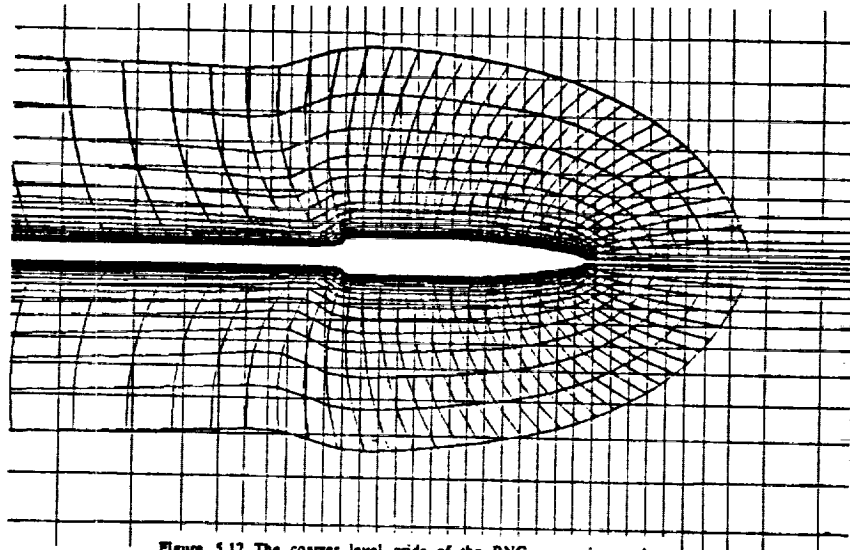


Figure 5.12 The coarser level grids of the BNC composite mesh

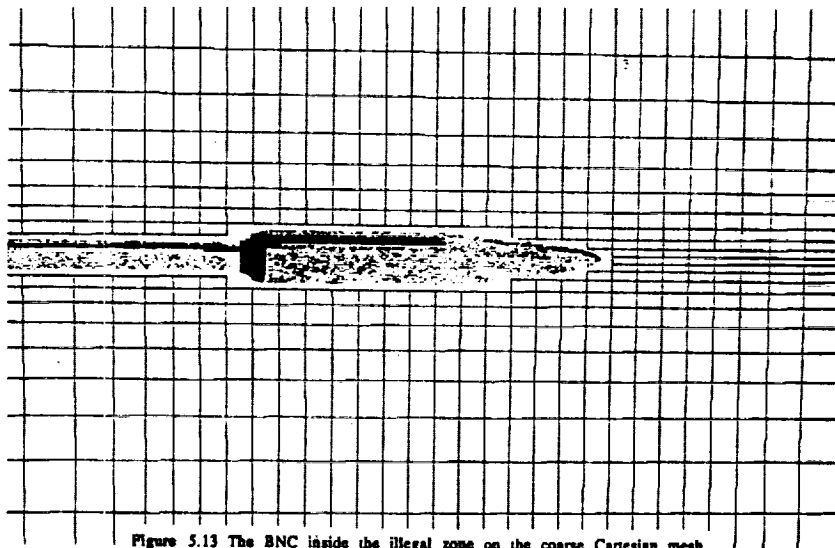


Figure 5.13 The BNC inside the illegal zone on the coarse Cartesian mesh

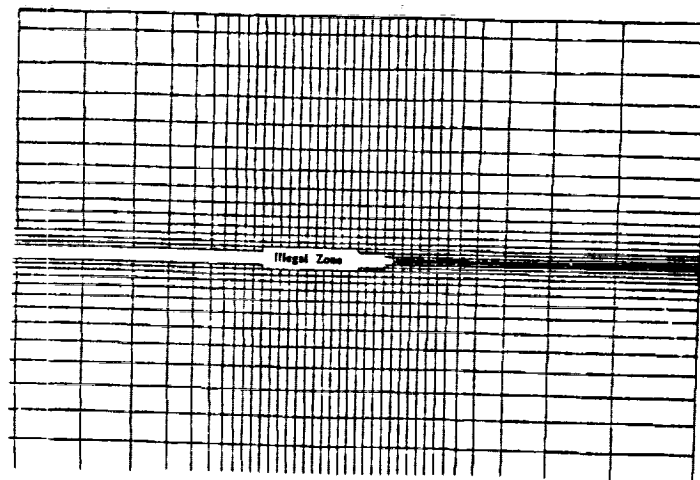


Figure 5.14 Illegal zone created in the coarse Cartesian mesh

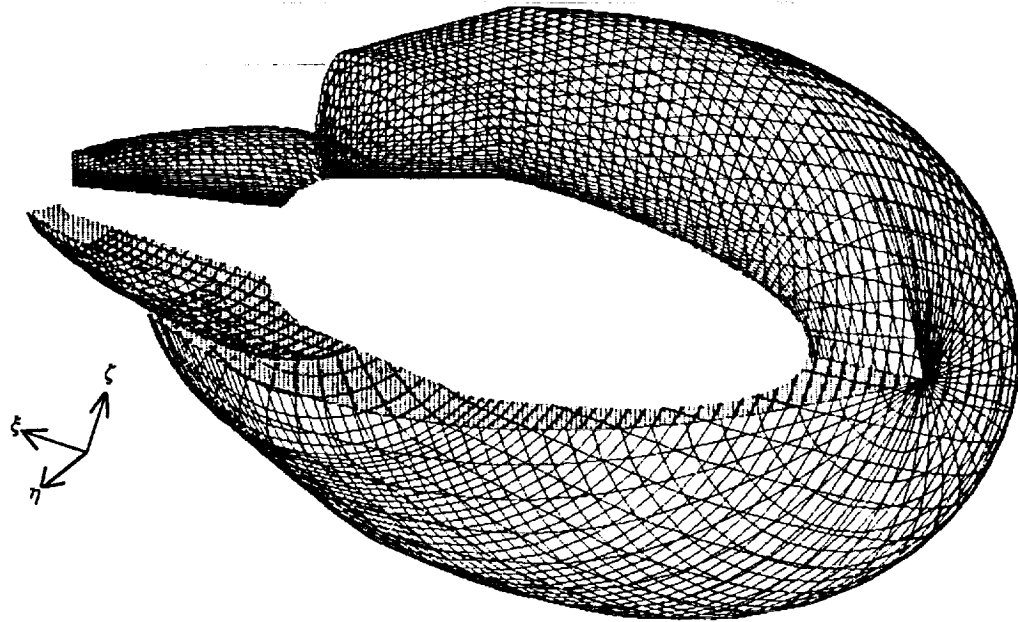


Figure 5.16 Inner shell BNC C-O grid of the simple overlapped case

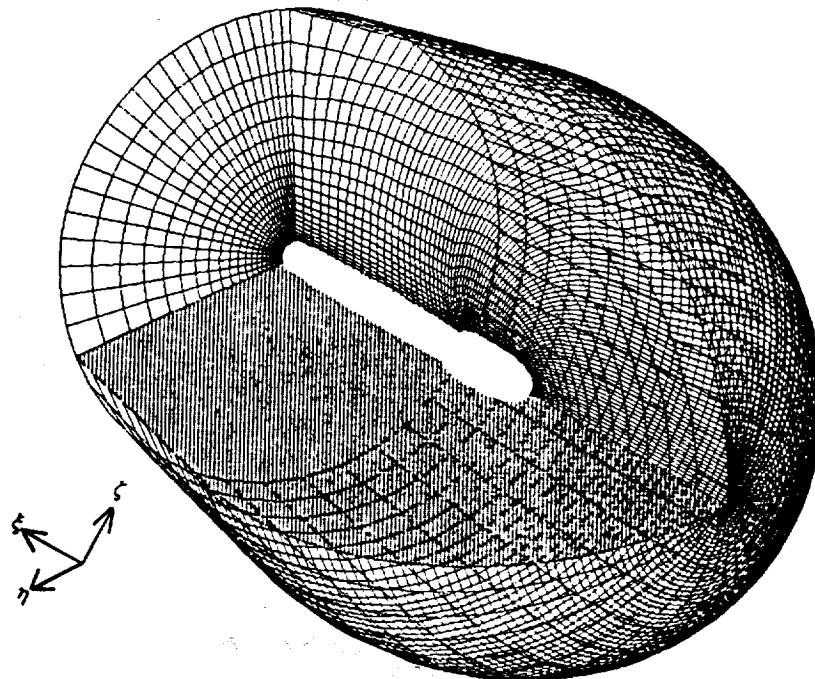


Figure 5.17 Outer shell BNC C-O grid of the simple overlapped case

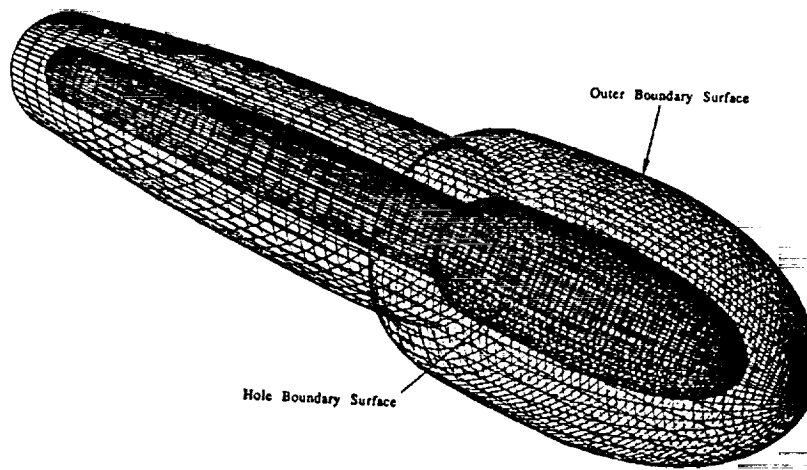


Figure 5.18 Hole boundary and outer boundary in the composite BNC simple overlapped case

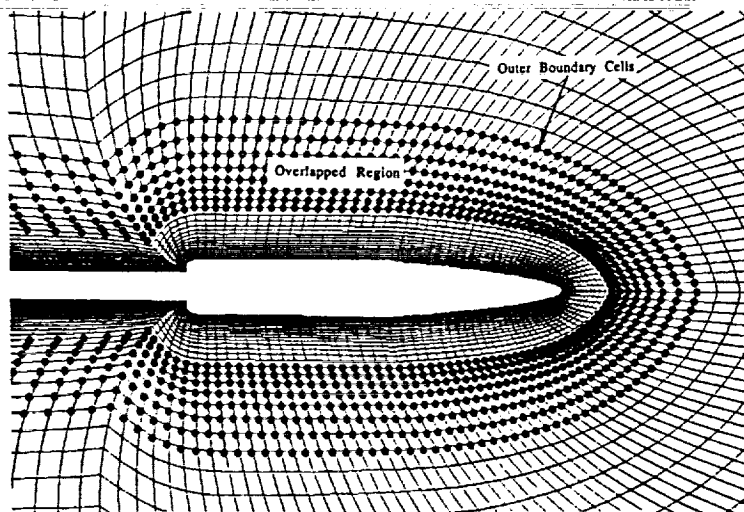


Figure 5.19 Overlapped region in the symmetry plane of the simple composite BNC case

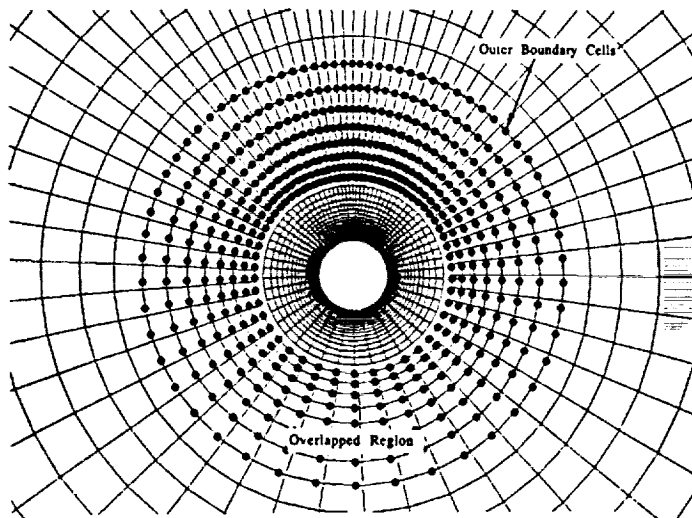


Figure 5.20 Overlapped region in the crossflow plane of the simple composite BNC case

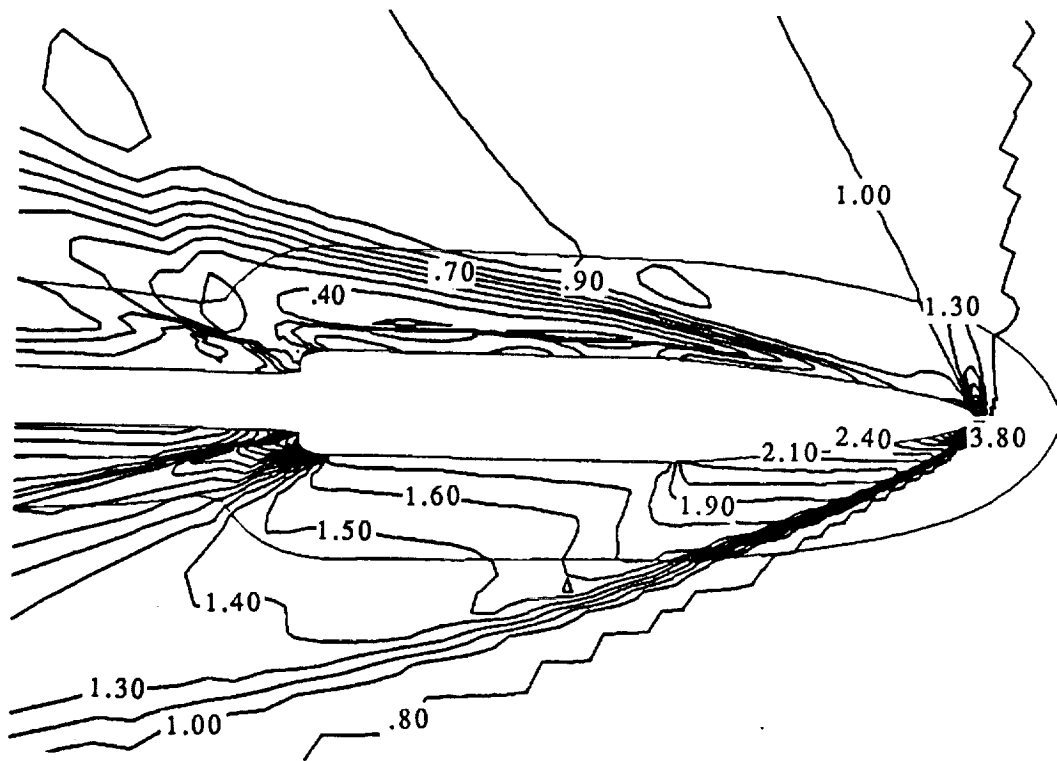


Figure 5.21 Normalized pressure contours on the longitudinal plane of the BNC simple case

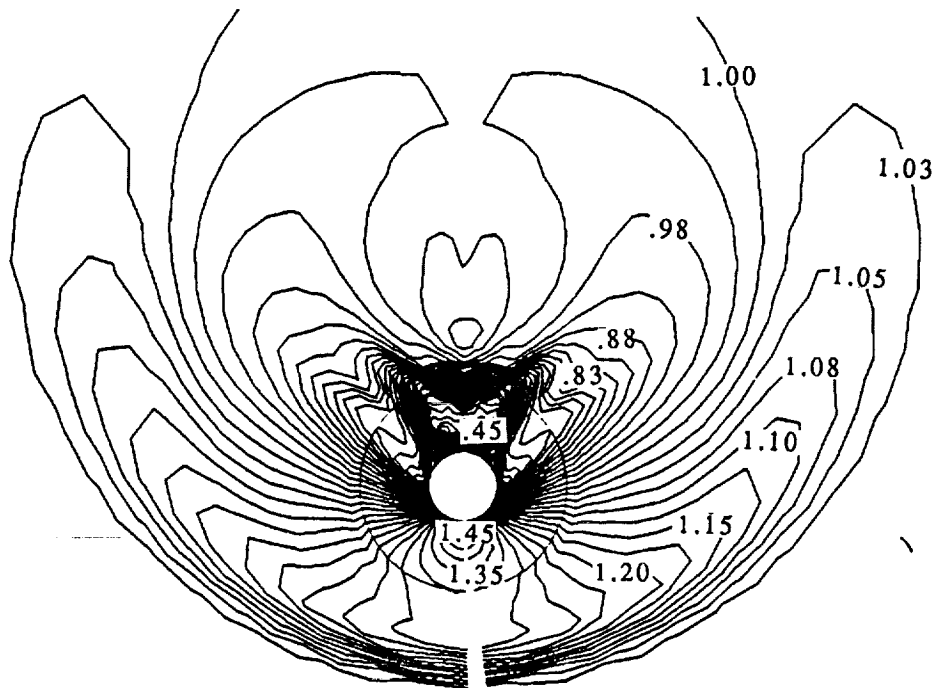
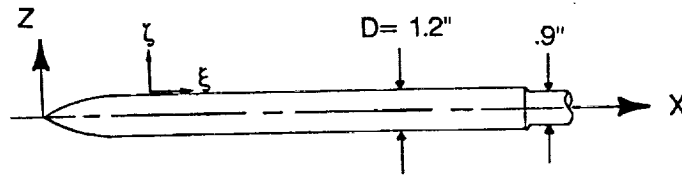


Figure 5.22 Normalized density contours on the crossflow plane of the BNC simple case



$$\begin{aligned} Z/D &= (5.26^2 - (X/D - 2.25)^2)^{1/2} - 4.76 & 0.00 < X/D < 2.25 \\ Z/D &= .500 & 2.25 < X/D < 18.00 \end{aligned}$$

Figure 5.23 Schematic of the ogive-nose cylinder (ONC)

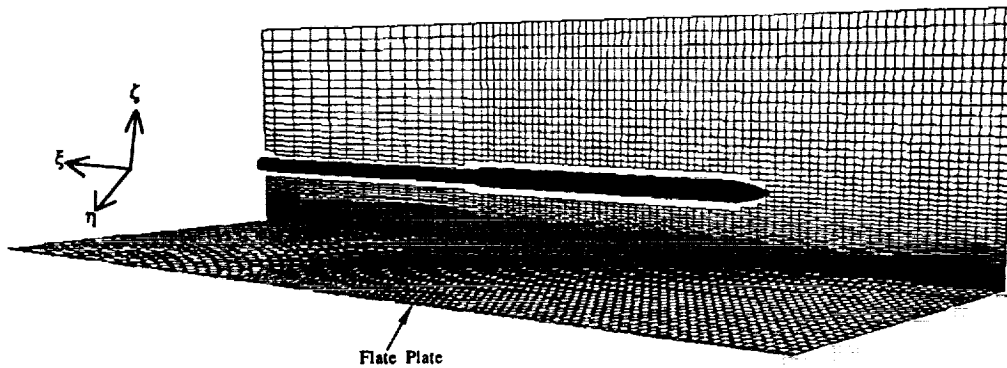


Figure 5.24 3-D view of the ogive-nose cylinder in a Cartesian mesh with an overlapped region

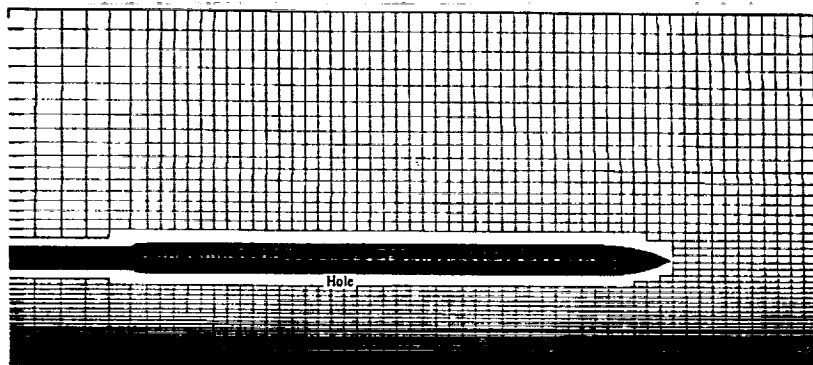


Figure 5.25 Symmetry plane composite view of the ogive-nose cylinder near a flat plate



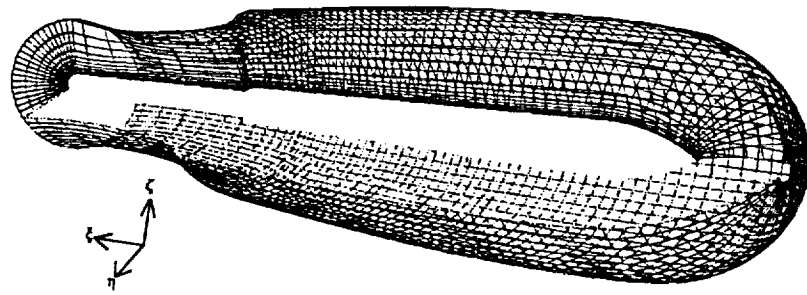


Figure 5.26 Ogive-nose cylinder C-O mesh

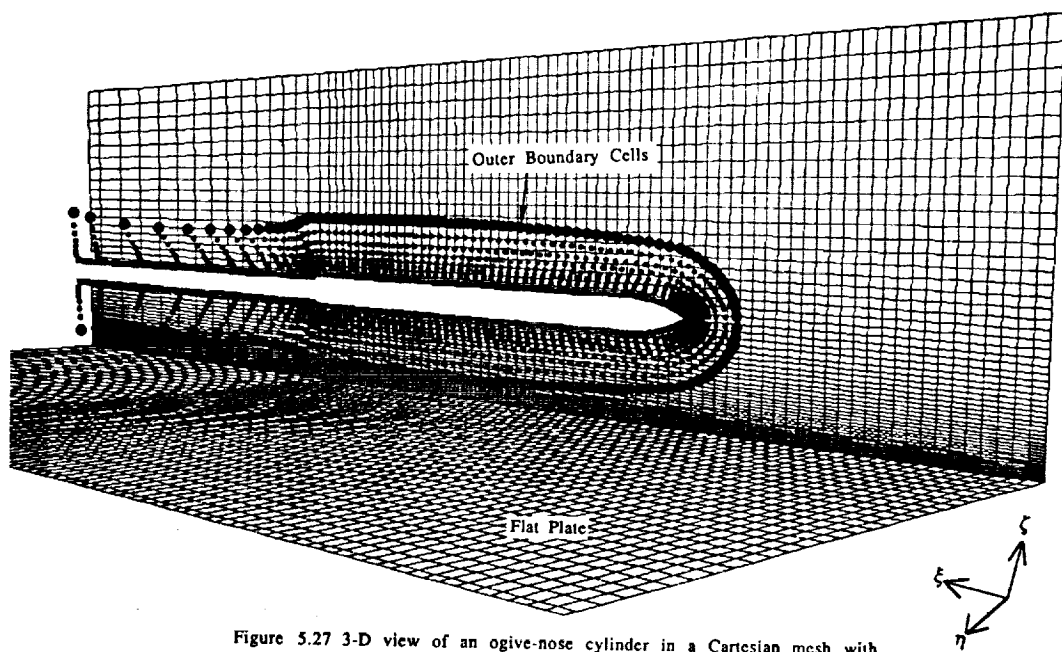


Figure 5.27 3-D view of an ogive-nose cylinder in a Cartesian mesh with overlapped region

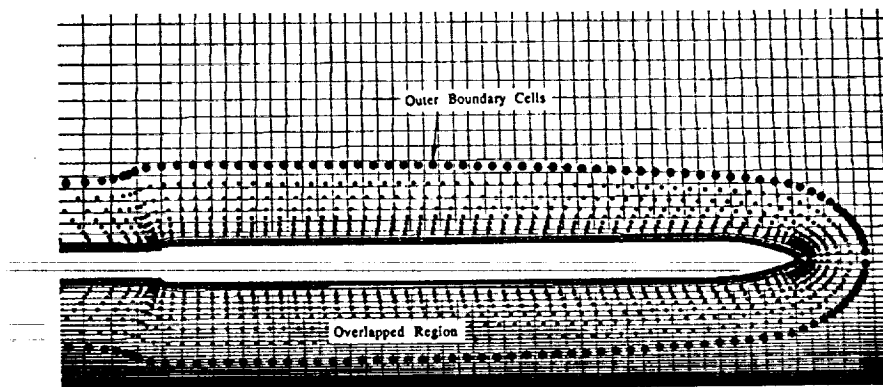


Figure 5.28 A detailed symmetry plane view of the composite ONC mesh with overlapped region

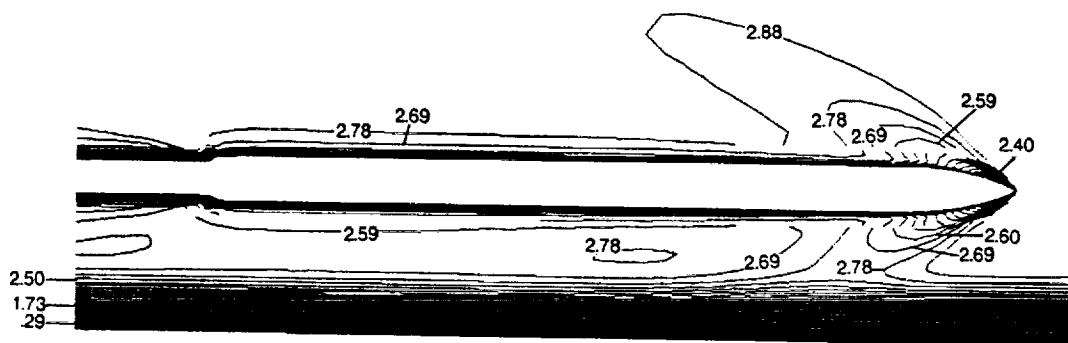


Figure 5.29 Mach number contours on the symmetry plane of the composite ONC which is at (3.5D) distance from the flat plate

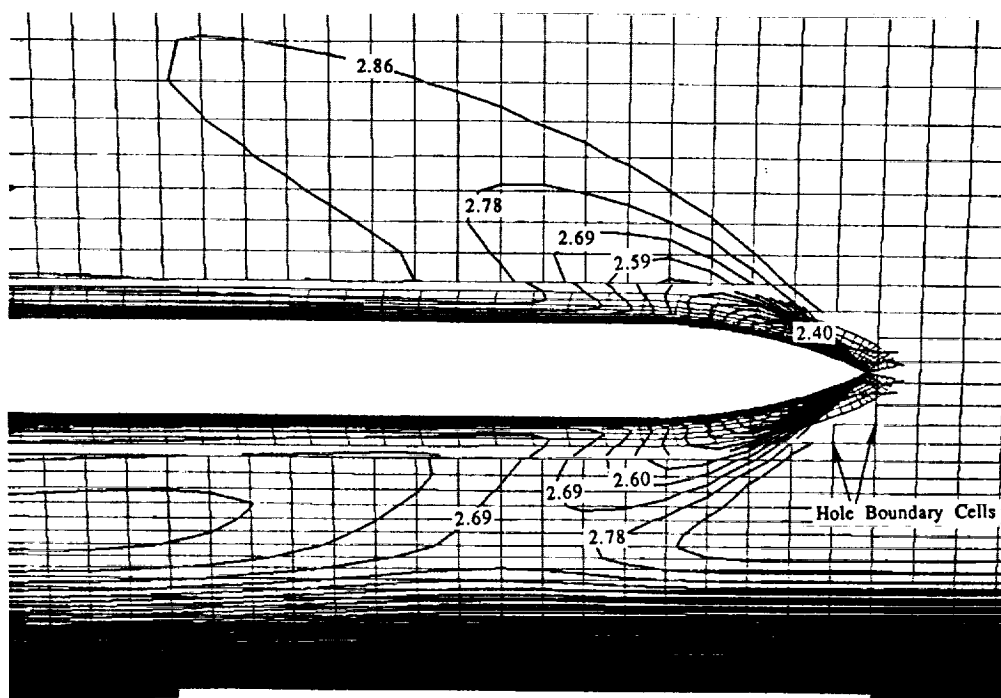


Figure 5.30 Detailed view of Mach contours on the symmetry plane of the composite ONC mesh

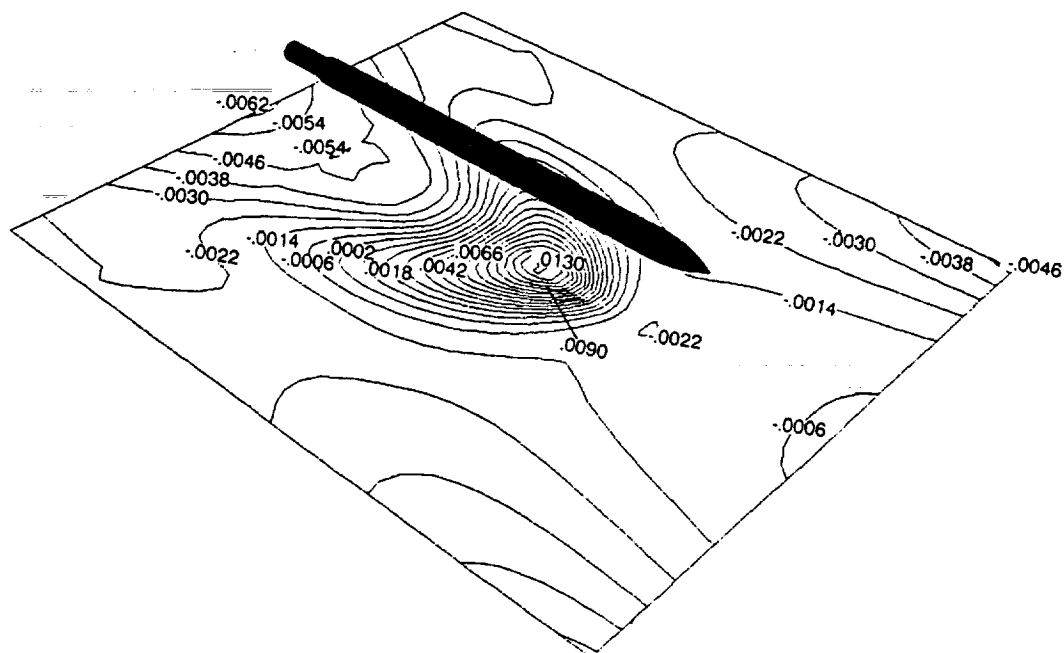


Figure 5.31 Pressure coefficient contours on the surface of the flat plate with the ONC at (3.5D) distance

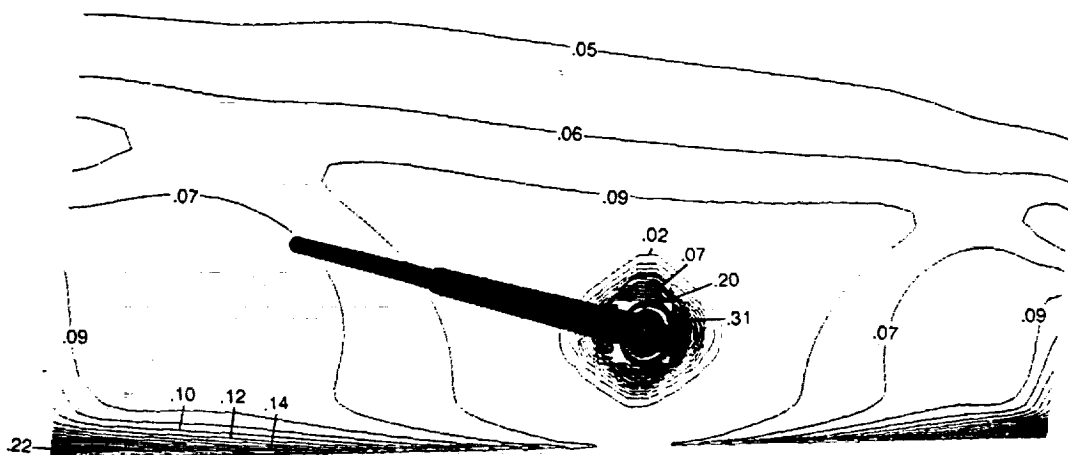


Figure 5.32 Crossflow Mach number contours at the forebody-cylinder junction of composite ONC

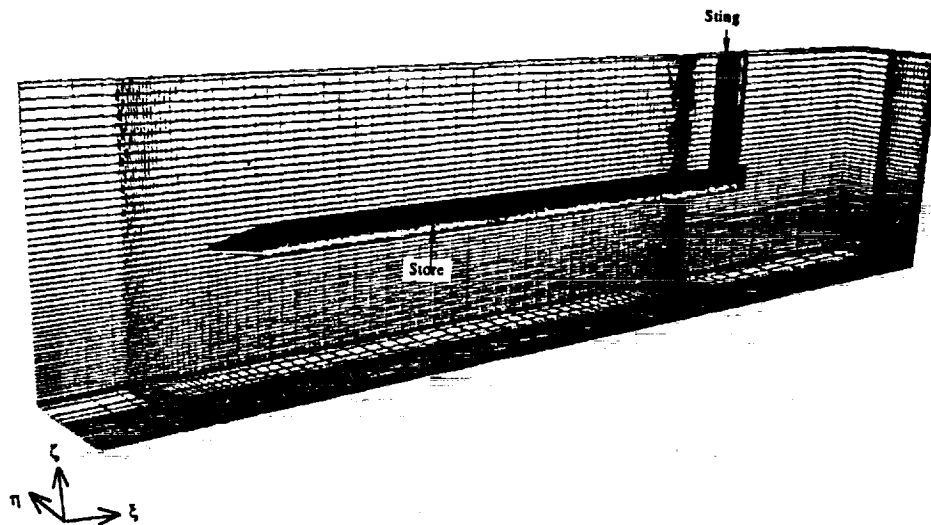
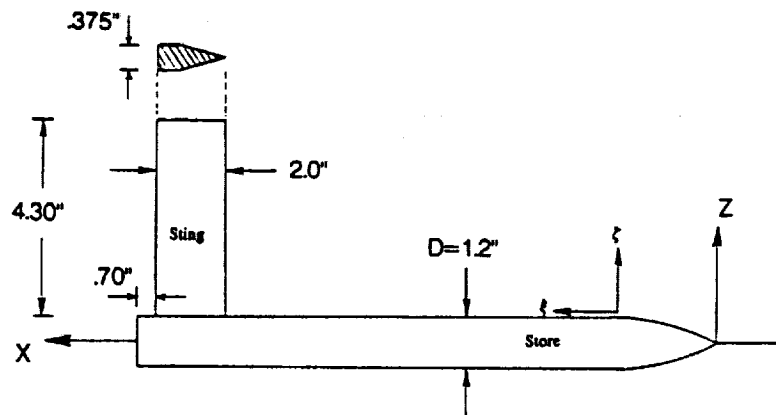


Figure 5.33 Ogive-nose store connected to an L-shaped sting embedded in a Cartesian farfield



$$Z/D = (9.58 \times 10^{-2} - (X/D - 3.06)^2)^{1/2} \quad 0 < X/D < 3.06$$

$$Z/D = .500 \quad 3.06 < X/D < 22.69$$

Figure 5.34 Schematic of the store cylinder connected to an L-shaped sting

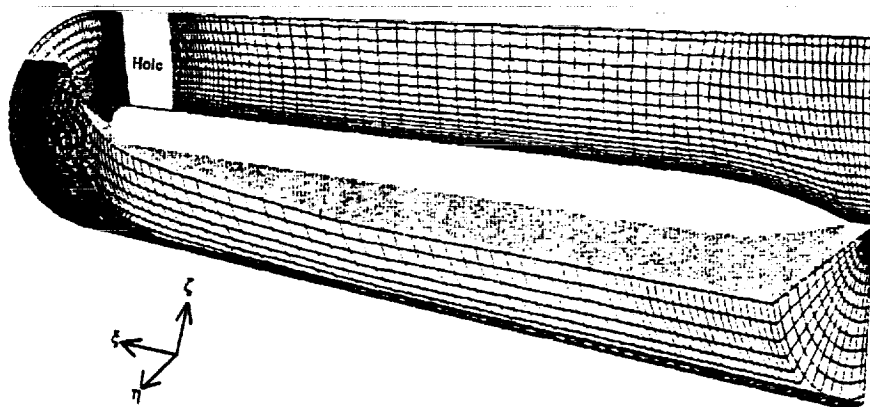


Figure 5.35 Ogive-nose store H-O grid which includes hole in the region of the sting

ORIGINAL PAGE IS  
OF POOR QUALITY

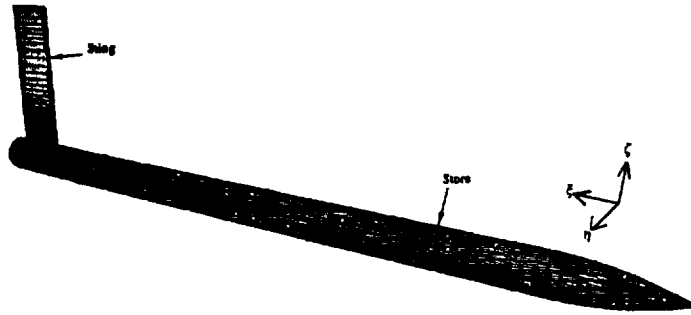


Figure 5.36 L-shaped sting connected to the surface of the store body.  
a) entire view of both surfaces  
b) detailed view at the base of the sting

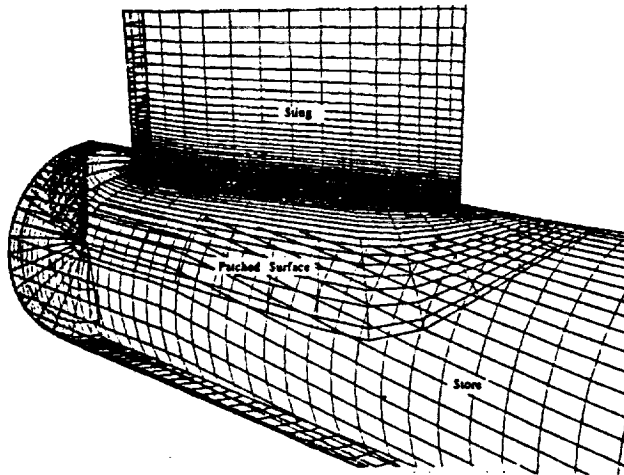


Figure 5.37 Cross sectional grid of the L-shaped sting at its base

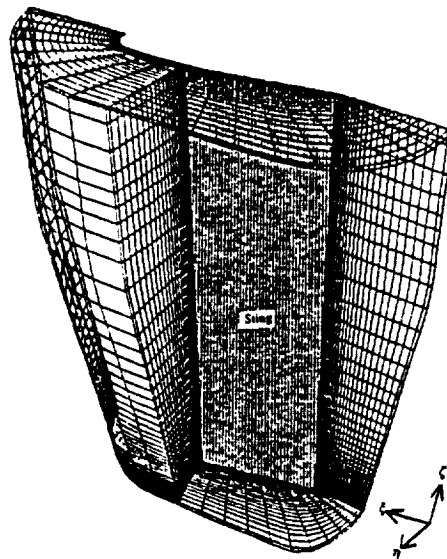


Figure 5.38 O-H grid of the L-shaped sting

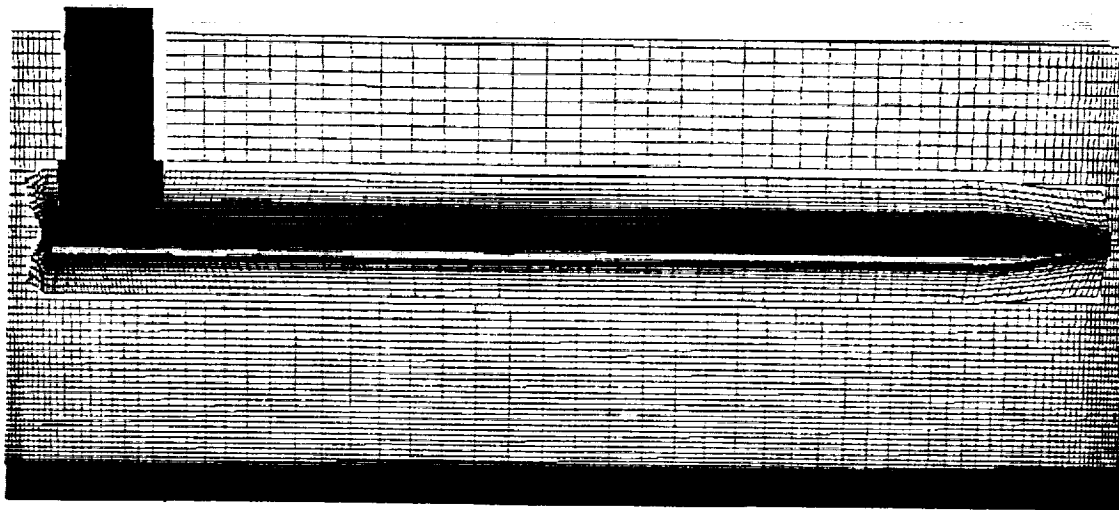


Figure 5.39 Composite store connected to the L-shaped sting in a Cartesian farfield mesh,  
a) symmetry view of the composite mesh

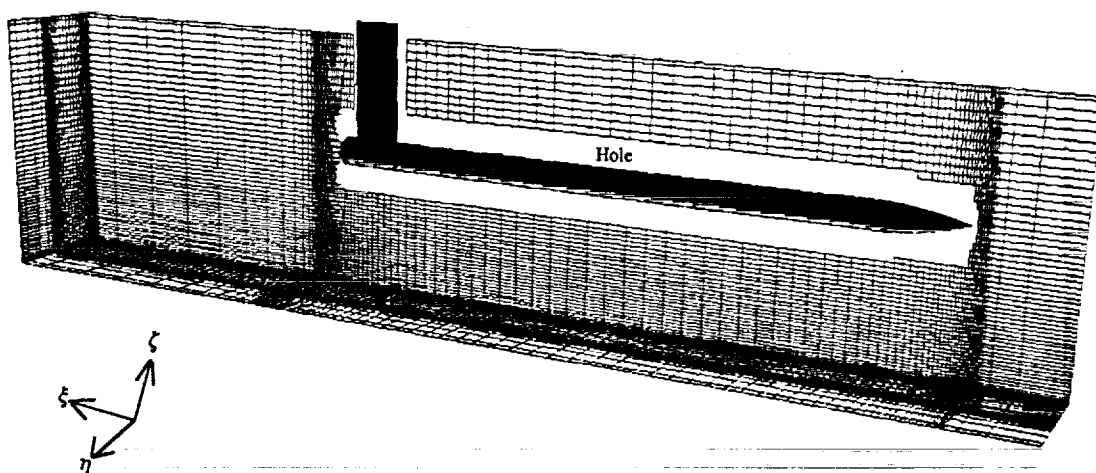


Figure 5.39 Composite store connected to the L-shaped sting in a Cartesian farfield mesh,  
b) holes generated in the cartesian mesh caused by the embedded grids

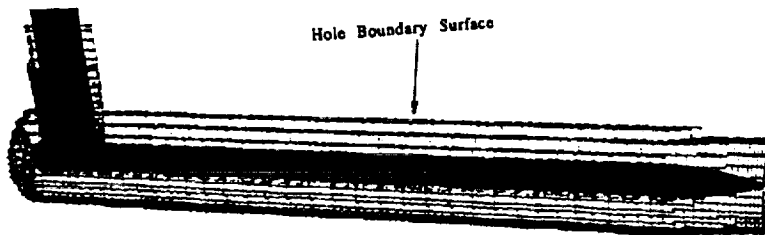


Figure 5.40 The hole boundary surface in the Cartesian mesh that surrounds the store and sting grids

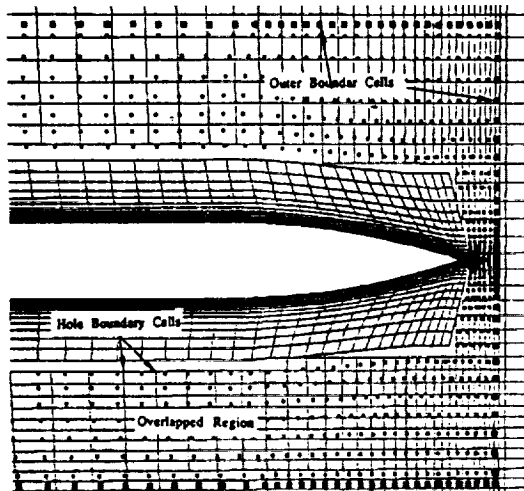


Figure 5.41 Overlapped region between the store, sting, and Cartesian grids.  
a) 8 cell overlapped region at the nose of the store  
b) 8 cell overlapped region at the base of the store

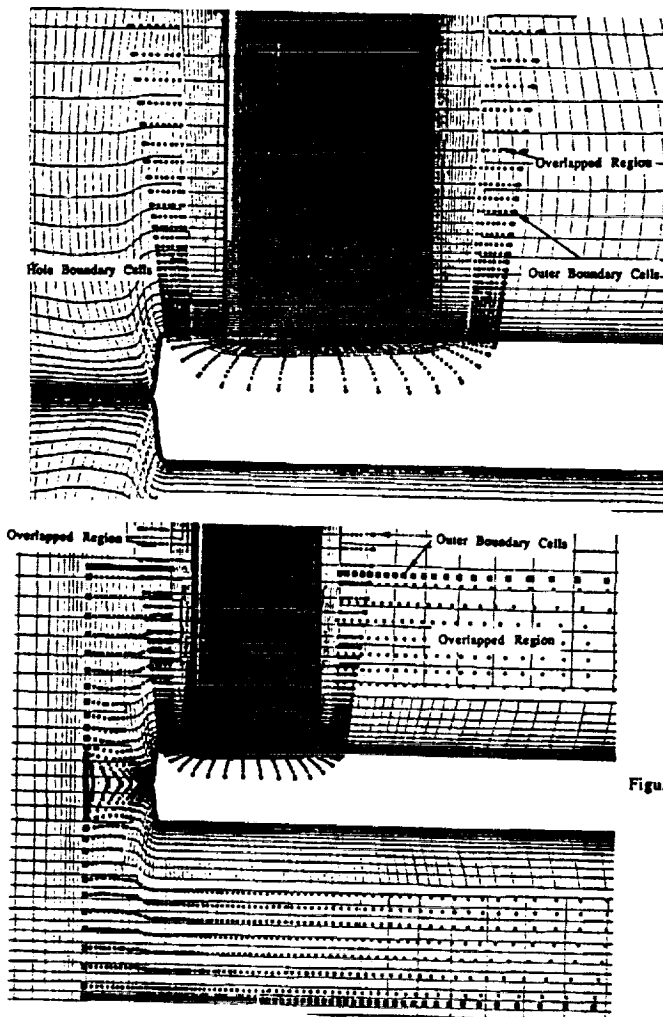


Figure 5.42 Detailed view of the overlapped region between the store and sting meshes

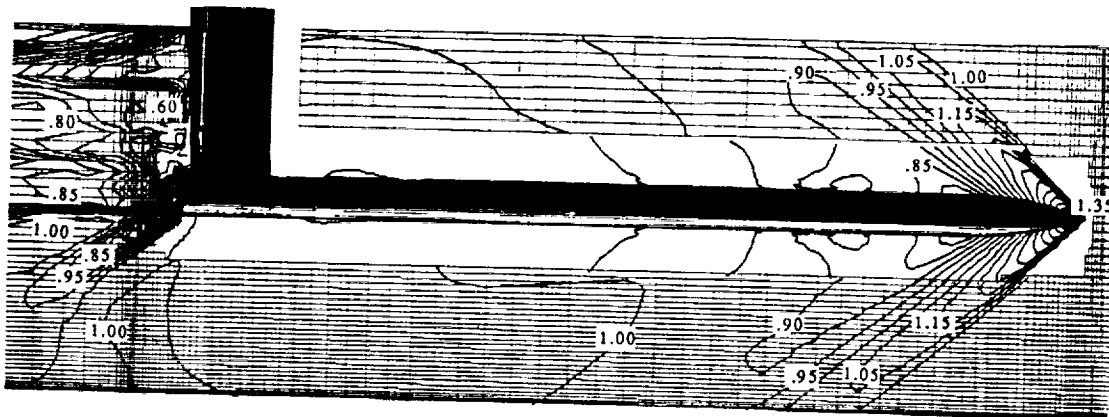


Figure 5.43 Symmetry plane normalized density contours of the composite store and sting embedded within a Cartesian farfield

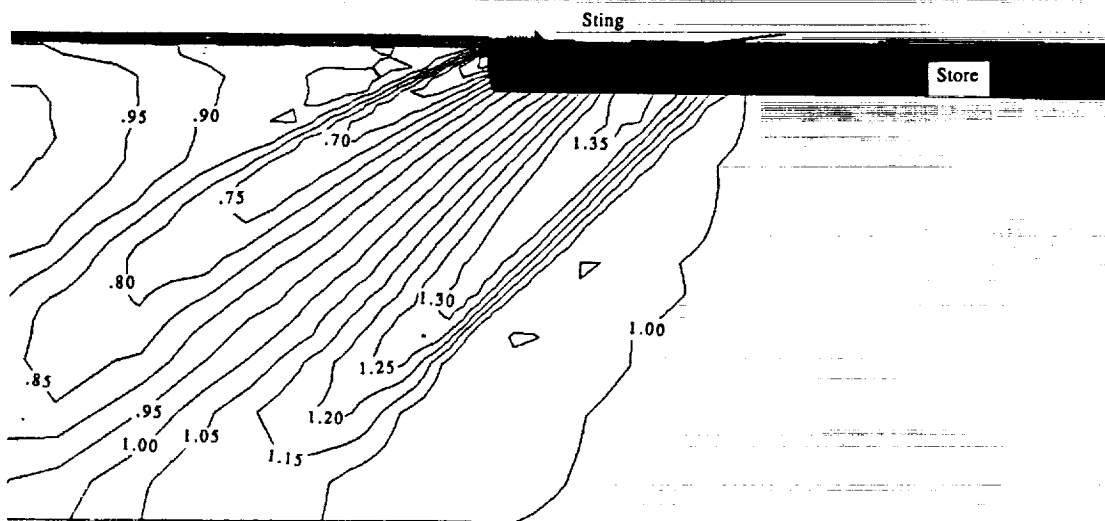


Figure 5.44 Top view of the longitudinal density contours around the sting in the composite mesh



ORIGINAL PAGE IS  
OF POOR QUALITY

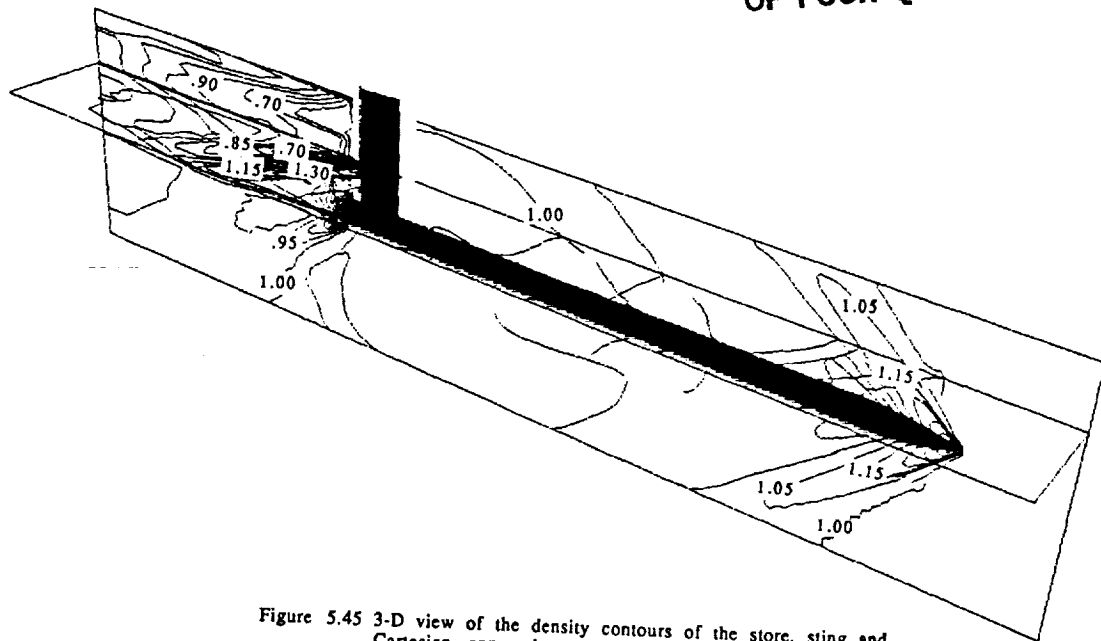


Figure 5.45 3-D view of the density contours of the store, sting and Cartesian composite mesh

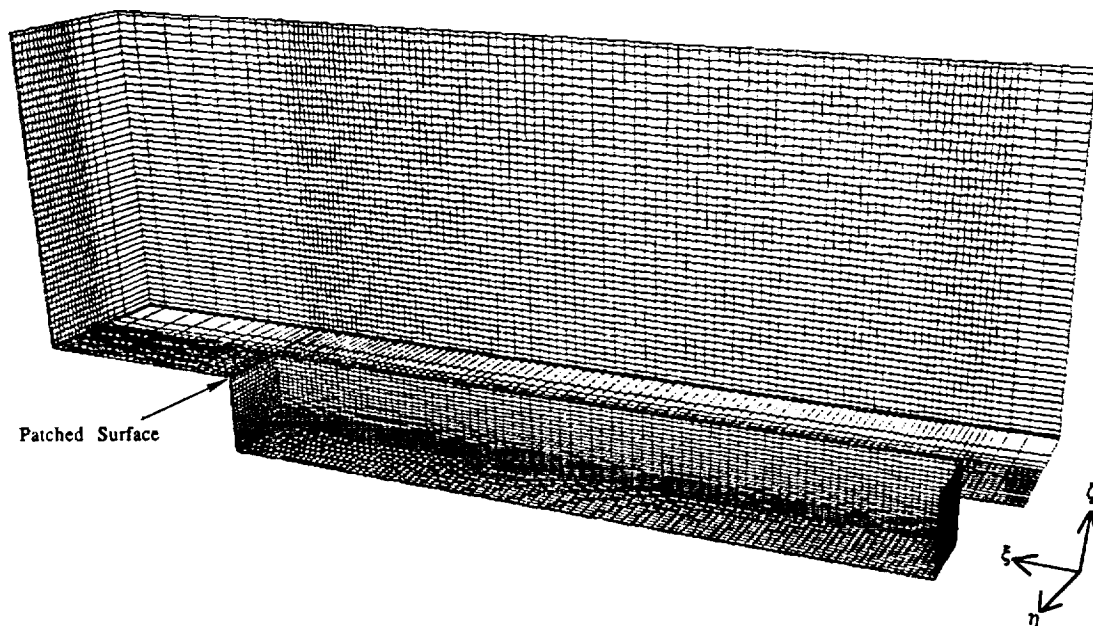


Figure 5.46 Rectangular cavity patched to the Cartesian farfield

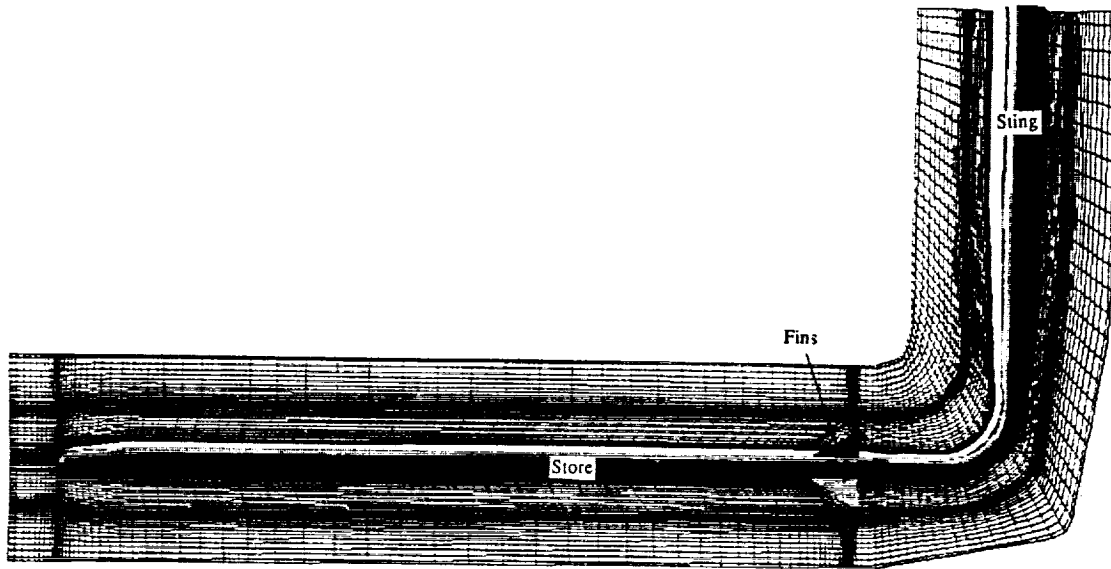


Figure 5.47 The combined three H-O grids generate the surface of the store with fins and curved sting

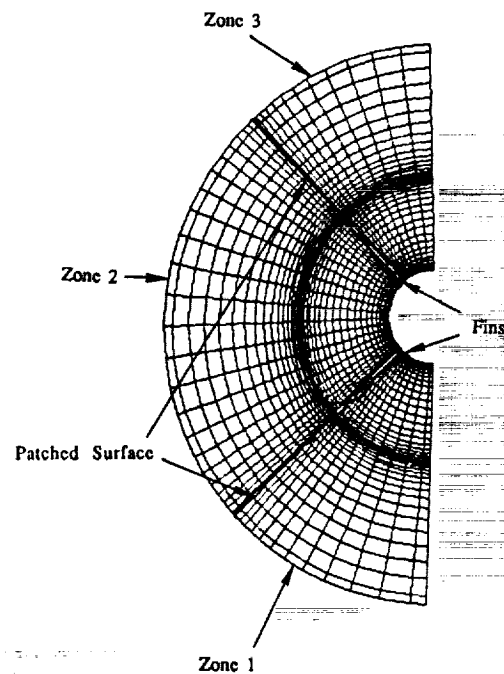


Figure 5.48 Cross sectional view of the three zonal grids that define the store

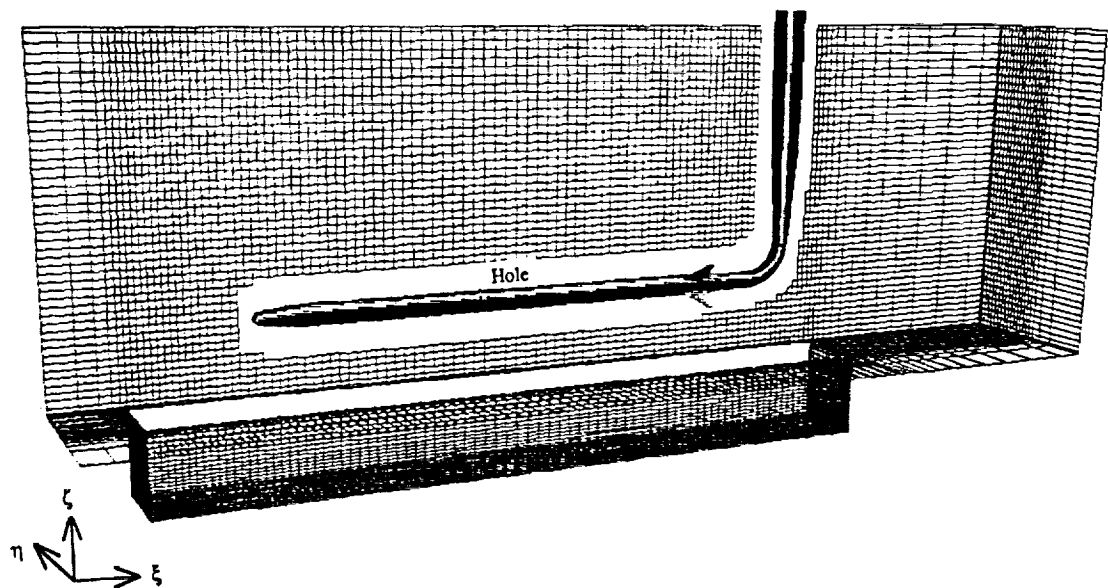


Figure 5.49 The hole generated in the farfield mesh by the three store grids

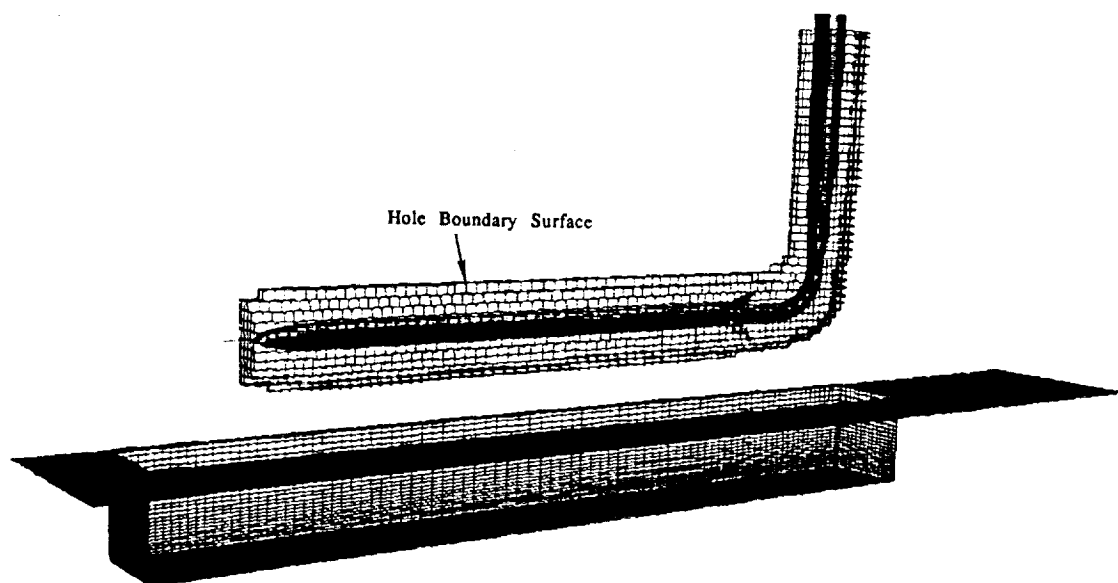


Figure 5.50 The hole boundary surface created in the Cartesian farfield mesh by the three zonal store grids

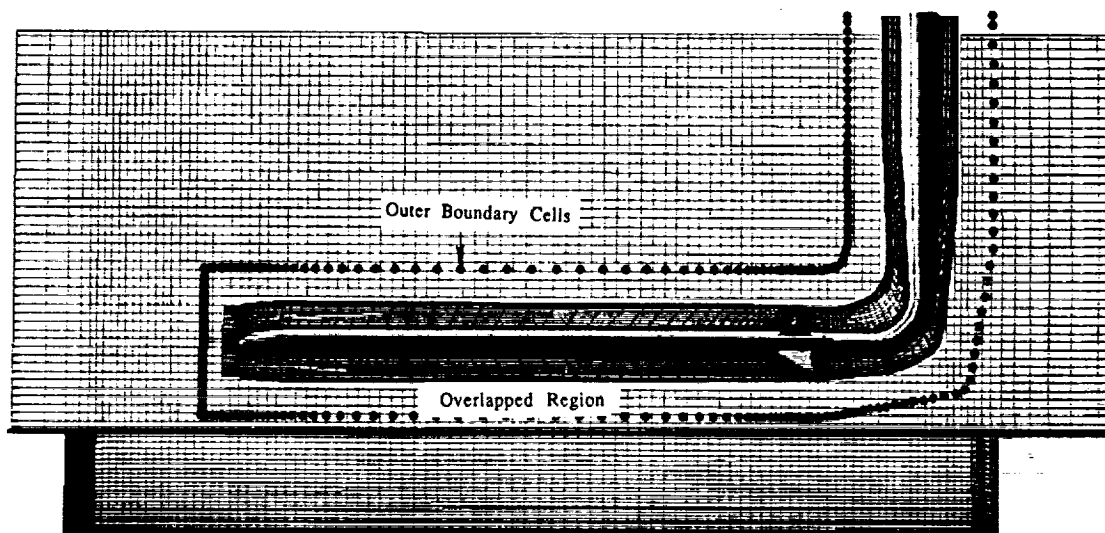


Figure 5.31 Symmetry view of the overlapped region between the store grids and the cartesian farfield mesh

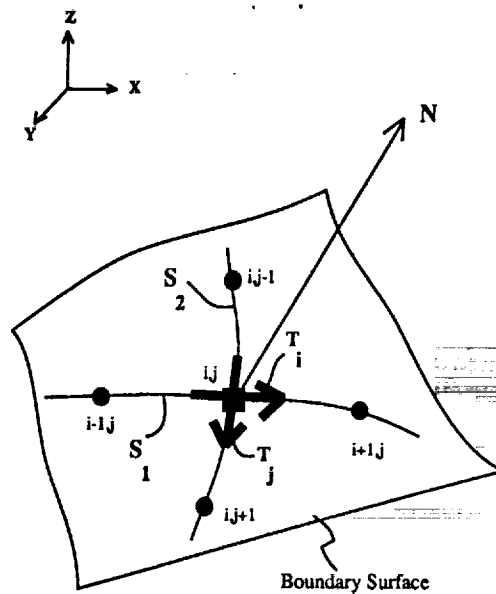


Figure A.1 Normal vector to a boundary surface

## APPENDIX A

### CALCULATIONS OF NORMAL VECTORS

The determination of a unit normal vector  $\vec{N}$ , to a hole boundary cell surface, used in locating hole cells, is explained in this appendix. The outward normal vector at a surface cell center is computed by constructing tangent vectors along grid lines, and performing their cross product. The procedure itself is simple, but care is needed so that the normal vectors are always pointing out of the surface. To obtain an outward normal vector, the hole boundary surface is defined in a counter-clockwise direction from 'i' to 'j' constant lines for a left-handed coordinate system, or clockwise for a right-handed system. A section of a boundary surface using a left-handed coordinate system is shown in Fig. A.1.

The arc lengths along a surface coordinate line are defined by

$$s_1 = \sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2} \quad (\text{A.1a})$$

$$s_2 = \sqrt{\Delta x_j^2 + \Delta y_j^2 + \Delta z_j^2} \quad (\text{A.1b})$$

and

$$\Delta x_i = x(i+1, j) - x(i-1, j) \quad (\text{A.2a})$$

$$\Delta x_j = x(i, j+1) - x(i, j-1) \quad (\text{A.2b})$$

The normalized tangents vectors,  $\vec{T}_i$  and  $\vec{T}_j$ , are defined by

$$\begin{aligned}\vec{T}_i &= T_{x_i} \hat{i} + T_{y_i} \hat{j} + T_{z_i} \hat{k} \\ \vec{T}_j &= T_{x_j} \hat{i} + T_{y_j} \hat{j} + T_{z_j} \hat{k}\end{aligned}\quad (A.3)$$

The components of the tangent vectors are defined by

$$\begin{aligned}T_{x_i} &= \frac{(x(i+1,j) - x(i-1,j))}{\Delta s_1} & T_{x_j} &= \frac{(x(i,j+1) - x(i,j-1))}{\Delta s_2} \\ T_{y_i} &= \frac{(y(i+1,j) - y(i-1,j))}{\Delta s_1} & T_{y_j} &= \frac{(y(i,j+1) - y(i,j-1))}{\Delta s_2} \\ T_{z_i} &= \frac{(z(i+1,j) - z(i-1,j))}{\Delta s_1} & T_{z_j} &= \frac{(z(i,j+1) - z(i,j-1))}{\Delta s_2}\end{aligned}\quad (A.4)$$

where

$$\begin{aligned}\Delta s_1 &= s_1(i+1,j) - s_1(i-1,j) \\ \Delta s_2 &= s_2(i,j+1) - s_2(i,j-1)\end{aligned}\quad (A.5)$$

The outward normal vector is then calculated by the cross product of the tangent vectors,

$$\begin{aligned}\vec{N} &= \vec{T}_i \times \vec{T}_j \\ \vec{N} &= N_x \hat{i} + N_y \hat{j} + N_z \hat{k}\end{aligned}\quad (A.6)$$

where

$$\begin{aligned}N_x &= T_{y_i} * T_{z_j} - T_{z_i} * T_{y_j} \\ N_y &= T_{z_i} * T_{x_j} - T_{x_i} * T_{z_j} \\ N_z &= T_{x_i} * T_{y_j} - T_{y_i} * T_{x_j}\end{aligned}\quad (A.7)$$

And, the unit normal vector is defined by

$$\hat{N} = \frac{\vec{N}}{|\vec{N}|} \quad \text{where} \quad |\vec{N}| = \sqrt{N_x^2 + N_y^2 + N_z^2} \quad (A.8)$$

Modifications in calculating the normal vectors are needed at the boundaries and the corners of the meshes. A two point extrapolation is used at the edges and an averaging type of extrapolation is used at the corners.

Extrapolation at the J-edge boundaries is given by

$$RI = \frac{(s_1(i,j) - s_1(i-1,j))}{(s_1(i-1,j) - s_1(i-2,j))}$$

$$RI1 = 1 + RI$$

$$Nx = Nx(i-1) * RI1 - Nx(i-2) * RI$$

$$Ny = Ny(i-1) * RI1 - Ny(i-2) * RI$$

$$Nz = Nz(i-1) * RI1 - Nz(i-2) * RI$$

(A.9)

Extrapolation at the I-edge boundaries is given by

$$RJ = \frac{(s_2(i,j) - s_2(i,j-1))}{(s_2(i,j-1) - s_2(i,j-2))}$$

$$RJ1 = 1 + RJ$$

$$Nx = Nx(j-1) * RJ1 - Nx(j-2) * RJ$$

$$Ny = Ny(j-1) * RJ1 - Ny(j-2) * RJ$$

$$Nz = Nz(j-1) * RJ1 - Nz(j-2) * RJ$$

(A.10)

Averaging extrapolation at the four corners is given by

$$RI = \frac{(s_1(i,j) - s_1(i-1,j-1))}{(s_1(i-1,j-1) - s_1(i-2,j-1))}$$

$$RJ = \frac{(s_2(i,j) - s_2(i-1,j-1))}{(s_2(i-1,j-1) - s_2(i-1,j-2))}$$

$$RIRJ = 1.0 + RI + RJ$$

$$Nx = Nx(i-1,j-1) * RIRJ - Nx(i-1,j-1) * RI - Nx(i-1,j-2) * RJ$$

$$Ny = Ny(i-1,j-1) * RIRJ - Ny(i-1,j-1) * RI - Ny(i-1,j-2) * RJ$$

$$Nz = Nz(i-1,j-1) * RIRJ - Nz(i-1,j-1) * RI - Nz(i-1,j-2) * RJ$$

(A.11)

## APPENDIX B

### JACOBIAN OF ISOPARAMETRIC TRANSFORMATION

The Jacobian matrix,  $M$ , and its inverse,  $M^{-1}$ , of the isoparametric transformation of a hexahedron in the physical space to a cube in the interpolation space is given below.

$$M = \begin{bmatrix} (a_2 + a_5\eta + a_6\zeta + a_8\eta\zeta) (a_3 + a_5\xi + a_7\zeta + a_8\xi\zeta) \\ (b_2 + b_5\eta + b_6\zeta + b_8\eta\zeta) (b_3 + b_5\xi + b_7\zeta + b_8\xi\zeta) \\ (c_2 + c_5\eta + c_6\zeta + c_8\eta\zeta) (c_3 + c_5\xi + c_7\zeta + c_8\xi\zeta) \\ (a_4 + a_6\xi + a_7\eta + a_8\zeta\eta) \\ (b_4 + b_6\xi + b_7\eta + b_8\zeta\eta) \\ (c_4 + c_6\xi + c_7\eta + c_8\zeta\eta) \end{bmatrix} \quad (B.1)$$

$M^{-1}$  exists as long as the mapping is one to one. Since  $M$  is a 3 by 3 matrix, its inverse can be computed as

$$M^{-1} = \begin{bmatrix} (M_{22}M_{33} - M_{23}M_{32}) & -(M_{12}M_{33} - M_{13}M_{32}) & (M_{12}M_{23} - M_{13}M_{22}) \\ -(M_{21}M_{33} - M_{23}M_{31}) & (M_{11}M_{33} - M_{13}M_{32}) & -(M_{11}M_{23} - M_{13}M_{21}) \\ (M_{21}M_{32} - M_{22}M_{31}) & -(M_{11}M_{32} - M_{12}M_{31}) & (M_{11}M_{22} - M_{12}M_{21}) \end{bmatrix} \frac{1}{\det M} \quad (B.2)$$

where

$$\det M = -(M_{11}M_{22}M_{33} + M_{12}M_{23}M_{31} + M_{13}M_{21}M_{32}) + (M_{13}M_{22}M_{31} + M_{12}M_{21}M_{33} + M_{11}M_{23}M_{32}) \quad (B.3)$$





## Report Documentation Page

1. Report No. NASA CR-182008		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle  <b>An Overlapped Grid Method for Multigrid, Finite Volume/ Difference Flow Solvers - MaGGiE</b>				5. Report Date February 1990	
				6. Performing Organization Code	
7. Author(s) Oktaý Baysal Victor R. Lessard				8. Performing Organization Report No.	
				10. Work Unit No. 505-68-91-01	
9. Performing Organization Name and Address Old Dominion University Research Foundation P.O. Box 6369 Norfolk, Virginia 23508-0369				11. Contract or Grant No. NAG1-664	
				13. Type of Report and Period Covered Contractor Report 1987-1989	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665				14. Sponsoring Agency Code	
15. Supplementary Notes Technical Monitor: David S. Miller, Langley Research Center Oktaý Baysal: Principal Investigator					
16. Abstract The objective of this work is to develop a domain decomposition method via overlapping/embedding the component grids, which is to be used by upwind, multi-grid, finite volume solution algorithms. A computer code, given the name MaGGiE, (Multi-Geometry Grid Embedder), is developed to meet this objective. MaGGiE takes independently generated component grids as input, and automatically constructs the composite mesh and interpolation data, which can be used by the finite volume solution methods with or without multigrid convergence acceleration. Six demonstrative examples, showing various aspects of the overlap technique are presented and discussed. These cases are used for developing the procedure for overlapping grids of different topologies, and to evaluate the grid connection and interpolation data for finite volume calculations on a composite mesh. Time fluxes are transferred between mesh interfaces using a trilinear interpolation procedure. Conservation losses are minimal at the interfaces using this method. The multi-grid solution algorithm, using the coarser grid connections, improves the convergence time history as compared to the solution on composite mesh without multigridding.					
17. Key Words (Suggested by Author(s)) Overlapped Grids, Domain Decomposition, Computational Fluids, Navier-Stokes Solutions, Complex Geometries, Missile Aerodynamics, Store Carriage and Separation, Cavity				18. Distribution Statement  Unclassified-unlimited  Subject Category 02	
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of pages 111	
				22. Price A06	

